

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E SISTEMAS.
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**Uma Arquitetura Baseada em Agentes
para Suporte ao Ensino à Distância**

**Tese submetida à Universidade Federal de Santa Catarina para
a obtenção do título de Doutor em Engenharia de Produção**



0.302.676-7

UFSC-BU

MARCELLO THIRY COMICHOLI DA COSTA

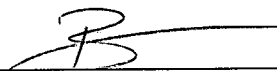
Florianópolis, abril de 1999



Uma Arquitetura Baseada em Agentes para Suporte ao Ensino à Distância

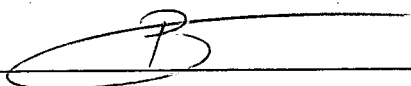
MARCELLO THIRY COMICOLI DA COSTA

Esta tese foi julgada adequada para a obtenção do título de DOUTOR EM ENGENHARIA DE PRODUÇÃO e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Produção.

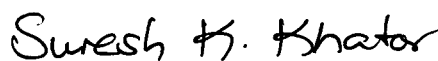


Prof. Ricardo de Miranda Barcia, Ph.D.
Coordenador do Curso

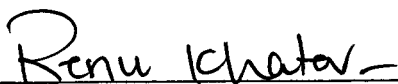
BANCA EXAMINADORA:



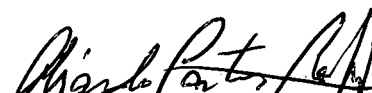
Prof. Ricardo de Miranda Barcia, Ph.D.
Orientador



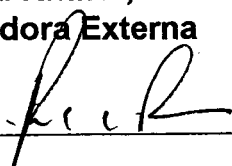
Prof. Suresh Khator, Ph.D.
Examinador Externo



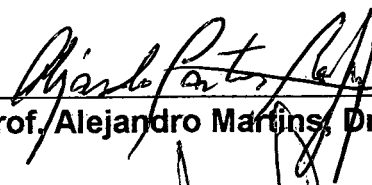
Prof.ª Renu Khator, Ph.D.
Examinadora Externa



Prof. Alejandro Martins, Dr. Eng.



Prof. Roberto Pacheco, Dr. Eng.



Prof. Ariovaldo Bolzan, Dr.
Moderador

À memória de meu avô,
Paul Ferdinand Thiry.

Agradecimentos

Em primeiro lugar, agradeço àquela luz maior que me amparou e conduziu durante todos os momentos dessa jornada, dando-me saúde e discernimento para superar as dificuldades, especialmente nos momentos de maior indecisão.

Agradeço, também, aos meus pais, Armando e Thereza, que com sua dedicação e carinho, me educaram e mostraram como trilhar meus próprios caminhos.

À Fernanda, pela sua eterna amizade e pela sua força constante, mesmo nos momentos mais difíceis.

Aos amigos Paulo, Celso, Luciano e Silvério do Núcleo de Computação da UNIVALI – São José, não apenas pelo apoio, mas também pelos momentos de descontração e alegria.

Ao pessoal da ACAFE, pelo apoio e pelos vários momentos de alegria em que tive a oportunidade de participar.

Ao professor Ricardo Barcia, por sua amizade, seu apoio e por toda a segurança e confiança transmitida para que eu pudesse executar este trabalho.

Aos professores Alejandro e Roberto, que procuram sempre me incentivar para a conclusão deste trabalho.

Aos professores Suresh Khator e Renu Khator da University of South Florida (USA), pela sua amizade e constante consideração demonstrada durante o período em que trabalhamos juntos. Agradeço também pela sua participação como examinadores externos e pelas sugestões recebidas para a melhoria deste trabalho.

A todos que direta ou indiretamente contribuíram para que eu pudesse realizar este trabalho.

À CAPES e à UNIVALI, por terem me proporcionado as condições para a realização deste trabalho.

Resumo

Existe atualmente uma demanda pela modelagem de sistemas de suporte ao ensino colaborativo em ambientes distribuídos. Estes sistemas são ambientes eletrônicos que suportam e mediam trabalho e aprendizado cooperativo em uma rede de computadores.

Este trabalho apresenta uma plataforma orientada a agentes para suportar a cooperação de usuários através da Internet. A plataforma proposta define uma arquitetura multi-agente que adota a estrutura de um sistema de federação de agentes. A comunicação dos agentes utiliza uma estrutura formalizada baseada na Linguagem de Comunicação de Agentes (ACL – *Agent Communication Language*).

A arquitetura distribuída é organizada através de três classes de agentes: agentes de interface, agentes de informação e agentes consultivos. Esta abordagem foi inicialmente proposta em (Thiry, 1998a). Uma extensão desta abordagem para a aplicação no ensino à distância foi proposta em (Thiry, 1998b). Os agentes utilizam técnicas diferentes para resolver problemas específicos e interpretar situações. As principais abordagens de raciocínio utilizadas são raciocínio baseado em casos (o enfoque é ensino baseado em casos) e regras de produção.

O trabalho discute a utilização da plataforma descrita como uma interessante ferramenta para suportar o ensino à distância, apresentando uma aplicação para seu uso através da Internet.

Abstract

Currently, there is a demand for modeling systems to support collaborative learning in distributed systems. These systems are electronic environments that support and mediate cooperative work and learning in a computer network.

This work introduces an agent-oriented framework to support the cooperation of users using Internet. The proposed framework defines a multi-agent architecture that adopts a federation of agents approach. The communication structure is based on the Agent Communication Language (ACL).

The distributed architecture is organized through three classes of agents: interface agents, information agents and advising agents. This approach was initially proposed in (Thiry, 1998a). An extension for the application in distance learning was proposed in (Thiry, 1998b). The agents use different techniques to solve specific problems and to interpret situations. The most important reasoning approach is based on cases (the focus is case-based teaching) and production rules.

This document discusses the use of the described framework as an interesting tool to support distance learning. An application is presented as an intelligent browser to Internet.

Sumário

1	<i>Introdução</i>	1
1.1	Origem do Trabalho	1
1.2	Objetivos do Trabalho	2
1.3	Justificativa do Trabalho	3
1.4	Estrutura do Trabalho	4
2	<i>Inteligência Artificial no Ensino</i>	6
2.1	Introdução	6
2.2	Histórico	6
2.3	Sistemas Tutores Inteligentes	7
2.4	Tendências Atuais da IA no Ensino	10
2.4.1	Novos Métodos de Ensino e Aprendizado	11
2.4.2	Aplicação de Agentes Inteligentes	12
2.4.3	Expandindo Métodos para Ensino e Aprendizado	13
2.5	Sistemas de Ensino Colaborativo	14
2.6	Colaboração Através da Internet	15
2.7	Conclusão	16
3	<i>Teoria de Agentes</i>	19
3.1	Introdução	19
3.2	Histórico	21
3.3	Definições de Agentes	25
3.4	Atributos dos Agentes	27
3.4.1	Autonomia	27
3.4.2	Mobilidade	28
3.4.3	Cooperação	28
3.4.4	Comunicabilidade	29
3.4.5	Aprendizagem	29
3.4.6	Reatividade	29
3.4.7	Habilidade Social	29

3.4.8	Pró-atividade	30
3.5	Tipologia de Agentes	30
3.5.1	Agentes Colaborativos	32
3.5.2	Agentes de Interface	33
3.5.3	Agentes Assistentes	34
3.5.4	Agentes de Recuperação de Informação	34
3.6	Comunicação entre Agentes	35
3.6.1	KQML	37
3.7	Arquiteturas de Agentes	39
3.7.1	Classificação de Arquiteturas	39
3.7.2	Arquitetura M	40
3.7.3	Arquitetura proposta por Genesereth	42
4	<i>Raciocínio Baseado em Casos</i>	45
4.1	Introdução	45
4.2	Histórico	47
4.3	Definição	48
4.4	A Abordagem RBC	50
4.4.1	Sistemas Interpretativos	52
4.4.2	Sistemas de Resolução de Problema	52
4.5	Representação do Conhecimento	53
4.6	Processo de Desenvolvimento RBC	53
4.6.1	Representação dos Casos	54
4.6.2	Indexação	56
4.6.3	Recuperação dos Casos	56
4.6.4	Adaptação	57
4.6.5	Aprendizagem	58
4.6.6	Análise da Situação	58
4.7	Ensino Baseado em Casos	59
5	<i>Modelo para Ensino Colaborativo</i>	62
5.1	Introdução	62
5.2	Definição do Modelo	63
5.3	Plataforma Multi-Agentes	64

5.3.1	Agente de Interface.....	65
5.3.2	Agente de Informação.....	67
5.3.3	Agente Consultivo	69
5.3.4	Arquitetura de Comunicação.....	70
5.3.5	Encapsulamento dos Casos	73
5.4	Aplicação no Ensino à Distância.....	74
5.4.1	Navegador Internet Inteligente.....	75
5.4.2	Aplicação em Engenharia de Software	78
6	<i>Conclusões e Perspectivas Futuras</i>	80
6.1	Conclusões.....	80
6.2	Perspectivas Futuras.....	81
7	<i>Referências Bibliográficas</i>.....	82

Lista de Figuras

Fig. 1: Interações entre os componentes de um sistema ITS	8
Fig. 2: Tipologia de Agentes proposta em (Nwana, 1996).....	31
Fig. 3: Taxonomia de Agentes proposta por Franklin e Graesser (1996).....	32
Fig. 4: Exemplo de uma mensagem KQML	38
Fig. 5: Arquitetura de Software M (Riecken, 1994).....	41
Fig. 6: Federação de agentes	43
Fig. 7: O Ciclo do RBC (Aamodt e Plaza, 1994).....	51
Fig. 8: Arquitetura genérica para o ambiente de ensino colaborativo	65
Fig. 9: Estrutura básica de agente de interface.....	66
Fig. 10: Agente de Informação.....	68
Fig. 11: Arquitetura de comunicação adotada	71
Fig. 12: <i>Performatives</i> KQML para conexão inicial entre os agentes e o Facilitador ...	72
Fig. 13: Arquitetura de comunicação com a inserção do navegador inteligente	76
Fig. 14: Ambiente colaborativo na forma de um navegador inteligente para Internet...	77
Fig. 15: Recuperação de um caso	78

1 Introdução

1.1 Origem do Trabalho

Apesar do ensino à distância não ser um tópico novo, ele vem ganhando uma maior atenção por parte da comunidade acadêmica. Este fato pode ser constatado pela crescente divulgação de programas de ensino à distância no Brasil e no mundo. Segundo o catálogo para 1998 da organização Peterson's (1997), apenas nos Estados Unidos existem mais de 700 instituições oferecendo cursos à distância. É importante notar que outra versão deste mesmo catálogo listava menos de 100 instituições em 1994. O surgimento de novas tecnologias de treinamento e educação, somadas aos requisitos de aprendizagem exigidos pela dinâmica da sociedade atual, fazem do ensino à distância uma real necessidade. O ponto principal é a possibilidade de expandir a sala de aula para um universo maior, permitindo educar pessoas em qualquer lugar e a qualquer hora. No entanto, Bork (1997) indica que um dos principais problemas com o aprendizado, incluindo-se aqui o ensino à distância, é a tendência em se confundir informação com aprendizado. Textos e leituras são primeiramente fontes de informação, em lugar de mídias de aprendizado. Além disso, a falta de interatividade existente entre os alunos usuários do programa de ensino à distância inviabiliza a colaboração, discussão e integração entre eles, comprometendo o processo de aprendizado como um todo. Assim sendo, é clara a necessidade de novas soluções que permitam realmente um ensino à distância.

Este trabalho adota o ensino à distância como um termo mais abrangente para educação e treinamento. De acordo com Porter (1997), educação implica em um conhecimento que pode ser altamente pragmático, envolvendo tanto os aspectos teóricos quanto os aspectos práticos, enquanto treinamento inclui o desenvolvimento de habilidades e conhecimento diretamente a partir de aplicações práticas.

Para que o ensino à distância possa existir, deve-se pensar em como será formada a sala de aula virtual. O ensino à distância pode incorporar várias tecnologias diferentes, as quais podem variar desde uma correspondência comum até os mais elaborados

programas de computador e meios de transmissão. Entretanto, uma das questões mais difíceis é definir quais tecnologias farão parte do ambiente a ser criado e qual a metodologia a ser empregada na integração destas.

Ainda segundo Porter (1997), uma sala de aula virtual não deveria ser muito diferente de uma real. Sendo assim, uma sala de aula deve possuir as seguintes características:

- Prover ferramentas de aprendizado disponíveis a medida que os alunos necessitam delas. É importante que estas ferramentas não sejam apresentadas todas de uma só vez ao aluno, mas que sejam facilmente localizadas.
- Possuir ambiente que cause uma expectativa para o aprendizado, colocando educador e aluno para compartilhar informações e trocar idéias.
- Permitir aos alunos a liberdade de experimentar, testar seu conhecimento, completar tarefas, e aplicar o que foi discutido ou lido.
- Oferecer mecanismos para avaliação do desempenho.

No planejamento de uma sala de aula virtual, seja quais forem as tecnologias envolvidas, deve ser considerado um ambiente que garanta as características supracitadas. Estas características sugerem um sistema colaborativo, onde os alunos possam interagir na discussão de idéias e aprender por experiência.

O contexto deste trabalho não procura abordar diretamente o planejamento de uma sala de aula virtual. A proposta é desenvolver um conjunto de recursos computacionais que permitam explorar as características definidas. Neste sentido, o presente trabalho oferece um suporte para a implantação de programas de ensino à distância através da Internet.

1.2 Objetivos do Trabalho

O objetivo geral desta pesquisa é propor um modelo de ambiente baseado em agentes inteligentes para dar suporte ao ensino à distância. A questão não é o desenvolvimento de uma aplicação de ensino à distância, mas a elaboração de um ambiente que possa ser utilizado como uma ferramenta de apoio durante o processo de aprendizado. Neste

sentido, este trabalho vem apresentar uma plataforma para o desenvolvimento de agentes de software que possibilite a interação e o trabalho cooperativo entre humanos e entre os próprios agentes. Uma proposta inicial para a aplicação desta plataforma no ensino à distância pode ser encontrada em (Thiry, 1998a).

Como objetivos específicos tem-se:

- Analisar a aplicação de agentes inteligentes e ensino baseado em casos em sistemas que permitam o trabalho cooperativo.
- Desenvolver uma plataforma para a construção de agentes inteligentes suportados por redes de computadores, utilizando uma linguagem de comunicação de agentes padronizada.
- Propor um modelo com agentes especializados, utilizando-se de técnicas de raciocínio baseado em casos e regras de produção.
- Implementar computacionalmente a arquitetura multi-agentes proposta.
- Apresentar a aplicação da arquitetura desenvolvida como ferramenta de apoio para o ensino à distância.

1.3 Justificativa do Trabalho

Para o sucesso de programas de ensino à distância através da Internet, deve-se considerar diferentes tipos de usuários, inclusive aqueles que possuem pouca ou nenhuma experiência em informática. Isto ressalta a necessidade de novas interfaces interativas entre homem e computador.

Ainda que os sistemas de navegação mais usuais permitam uma forma não linear de exploração da informação, não há nenhum tipo de preocupação com uma orientação sobre o conteúdo apresentado. Apesar da introdução de linguagens baseadas em *scripts* adicionarem novas capacidades de interatividade, elas ainda não são muito adequadas para soluções onde o aprendizado está envolvido. Dentro deste contexto, linguagens multiplataforma como Java podem se tornar um importante instrumento para construção de novos paradigmas de utilização da Internet.

Entretanto, mais importante do que o meio para se alcançar interatividade é a modelagem de ambientes integrados que venham oferecer diversas facilidades para os usuários envolvidos. Estas facilidades podem variar da simples capacidade de filtragem de correio eletrônico até orientação educacional oferecida por agentes de software.

Segundo Crook (1994), há uma evidente necessidade pelo desenvolvimento de sistemas que permitam o ensino colaborativo em ambientes distribuídos. Sistemas de ensino colaborativo são ambientes eletrônicos que suportam e mediam tanto o trabalho quanto o ensino cooperativo através de uma rede de computadores. Neste sentido, a aplicação de agentes inteligentes parece ser evidente, uma vez que agentes especializados podem funcionar como procuradores de confiança dos seus usuários, neste caso os alunos, dentro da Internet representando seus conhecimentos e interagindo com outros tipos de agente, tais como assistentes pessoais. Esta forma de organização permite que a cooperação seja um processo assíncrono e que o conhecimento do grupo esteja sempre disponível a todos os integrantes do grupo.

A aplicação de raciocínio baseado em casos no ensino tem obtido resultados interessantes. Trabalhos como os de Schank et al (1994), Ferguson et al (1992) e Burke e Kass (1996) são exemplos da utilização de casos como uma forma para apresentar situações relevantes aos alunos e incentivar a busca por soluções. Neste sentido, surge o tema ensino baseado em casos que explora tanto a capacidade básica de estudantes aprenderem a partir de histórias, bem como o interesse dos professores de encapsularem seu conhecimento através da apresentação de situações.

A integração de agentes com ensino baseado em casos é uma abordagem que procura unir os benefícios da assistência personalizada ao usuário, da capacidade de testar seus conhecimentos, de manter informações relativas às suas crenças, necessidades e hábitos, com o estabelecimento de ambientes que proporcionem o aprendizado cooperativo a partir da troca de experiências entre os usuários.

1.4 Estrutura do Trabalho

O presente trabalho encontra-se estruturado em seis capítulos.

Este capítulo descreve a motivação e os objetivos da pesquisa. Seu caráter é introdutório, sendo também ressaltada a justificativa.

O próximo capítulo aborda uma discussão sobre a aplicação da Inteligência Artificial no processo de aprendizagem. São levantados alguns questionamentos sobre as abordagens tradicionais e como ambientes colaborativos podem garantir interessantes aspectos pedagógicos.

O capítulo 3 aborda uma revisão sobre a teoria de agentes inteligentes. O termo agente é discutido em suas características básicas, sendo apresentadas algumas variações de interpretação. Os tipos de agentes e arquiteturas que formam a base deste trabalho são enfatizados.

No capítulo 4 revisa-se a literatura correspondente ao raciocínio baseado em casos, onde são apresentadas as etapas que compõe este processo de raciocínio. Procura-se também mostrar como esta técnica pode ser utilizada no processo educacional.

O capítulo 5 apresenta o modelo para ensino colaborativo, baseado em uma arquitetura multi-agentes. Suas funções e características são estabelecidas, juntamente com os aspectos que fazem deste modelo uma interessante ferramenta para o ensino.

O último capítulo apresenta as conclusões obtidas e mostra algumas possibilidades de ampliação do modelo.

Finalmente, é apresentada a bibliografia utilizada, bem como, a citada neste trabalho.

2 Inteligência Artificial no Ensino

2.1 Introdução

O computador tem sido utilizado na educação durante os últimos 20 anos, demonstrando ser um grande auxílio no processo de ensino/aprendizagem (Beck et al, 1998), (Urban-Lurain, 1998). Este capítulo procura apresentar como a inteligência artificial (IA) está contribuindo com novas abordagens, ao permitir a representação de algumas habilidades de raciocínio e conhecimento especialista voltadas ao ensino e aprendizado.

2.2 Histórico

Os primeiros sistemas voltados para o ensino através do computador foram o treinamento baseado em computador (CBT – *Computer-Based Training*) e instrução baseada em computador (CAI – *Computer Assisted Instructional*) (McArthur et al, 1993), (Beck et al, 1998). Usualmente, estes sistemas geravam conjuntos de problemas projetados para aumentar o desempenho do estudante em domínios baseados em habilidades, como aritmética e recuperação de vocabulário (Urban-Lurain, 1998). Nestes sistemas, a instrução não era individualizada para as necessidades do estudante. A proposta era apresentar um problema ao estudante, registrar a resposta e avaliar seu desempenho. Segundo Beck et al (1998), as decisões sobre como o estudante deveria navegar através do material era baseada em árvores de decisão. A sequência de perguntas e respostas era dirigida pelos acertos e erros dos estudantes, não sendo consideradas suas habilidades individuais.

Nos sistemas CBT e CAI, a maior parte do esforço de desenvolvimento estava dirigida a resolver problemas com o desempenho do tempo de resposta. Eles consideravam que se a informação fosse apresentada ao estudante, ele poderia absorvê-la (Urban-Lurain, 1998). Enquanto CBT e CAI podiam ser parcialmente efetivos em ajudar os estudantes, eles não forneciam o mesmo tipo de atenção individualizada que podia ser oferecido por um tutor humano. Segundo McArthur et al (1993), esta atenção poderia ser alcançada

através de um sistema que pudesse raciocinar sobre o domínio e sobre o estudante. Neste sentido, surgiu uma nova proposta de sistema: os sistemas tutores inteligentes (ITS – *Intelligent Tutoring Systems*).

2.3 Sistemas Tutores Inteligentes

Segundo Hall (1990), os sistemas tutores inteligentes são uma composição de diversas disciplinas como psicologia, ciência cognitiva e inteligência artificial. O objetivo principal destes sistemas é a modelagem e representação do conhecimento especialista humano para auxiliar o estudante através de um processo interativo.

Sistemas ITS oferecem considerável flexibilidade na apresentação do material e uma maior habilidade para responder às necessidades do usuário. Eles procuram não apenas ensinar, mas como ensinar, aprendendo informações relevantes sobre o estudante, proporcionando um aprendizado individualizado. Estes sistemas alcançam sua “inteligência” pela representação de decisões pedagógicas sobre como transmitir o material (ensinar), além de informações sobre o estudante. Isto permite uma grande interatividade do sistema com o estudante. Sistemas tutores inteligentes têm sido apresentados como altamente eficientes para a melhora do desempenho e motivação dos estudantes.

Existem diversas aplicações para sistemas ITS, nos mais variados domínios do conhecimento. Um interessante exemplo da aplicação de sistemas ITS pode ser encontrado em (Koedinger e Anderson, 1995). Eles apresentam um tutor inteligente para a disciplina de álgebra chamado PAT (*Practical Algebra Tutor*). Neste trabalho, eles reportam os resultados positivos da aplicação de sistemas tutores em larga escala nas escolas de segundo grau. Em (Azevedo e Tavares, 1998), é descrito um sistema ITS para o ensino de orientação a objetos. Mitrovic (1998) utiliza um tutor inteligente para o ensino da linguagem SQL (*Structured Query Language*).

É importante notar que a implementação de sistemas ITS trouxeram a tona outras questões pedagógicas. Segundo McArthur et al (1993), as tecnologias que permitem automatizar métodos tradicionais de ensino e aprendizagem estão também ajudando na criação de novos métodos e redefinindo as metas educacionais. Isto traz algumas

dificuldades iniciais, pois os métodos tradicionais de ensino são bem conhecidos e bem definidos, mas os novos métodos precisam ainda ser mais discutidos. Dentre estes novos métodos, poder-se-ia citar aspectos de colaboração, aprendizado por experiências ou visualização.

Neste contexto, surgem os ambientes de ensino interativo (ILE – *Interactive Learning Environment*) que são uma evolução natural dos sistemas ITS, onde se procura endereçar os novos métodos educacionais. McArthur et al (1993) acredita que a migração de sistemas ITS para ILE é um processo que representa um padrão na educação atual. O objetivo não é apenas ensinar as habilidades tradicionais de forma mais rápida, eficiente e com menos custo. O foco é trabalhar na mudança dos métodos educacionais para redefinir novas metas e aplicá-las também em sala de aula.

Os sistemas tutores inteligentes são usualmente definidos através de componentes. Woolf (1992) identificou quatro componentes principais: módulo estudante, módulo pedagógico, módulo domínio do conhecimento e módulo de comunicação. Em seu trabalho, Woolf considerava que o conhecimento especialista estava representado pelo módulo do domínio. Entretanto, em um trabalho mais recente, Beck et al (1998) subdividiram este módulo e definiram um quinto elemento: o módulo especialista. A figura 1 apresenta como estes módulos estão relacionados.

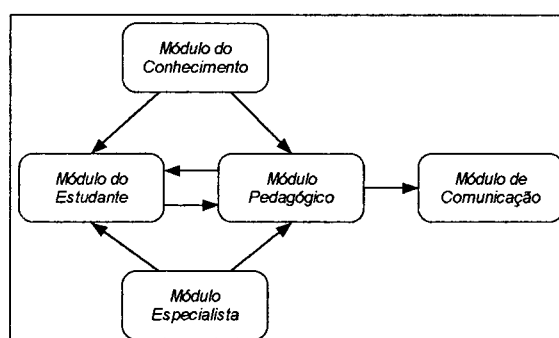


Fig. 1 – Interações entre os componentes de um sistema ITS.

O módulo estudante armazena informações específicas para cada estudante de forma individual. No mínimo, este módulo deve manter um histórico sobre como o estudante está trabalhando no material em questão. É interessante também manter registro sobre os erros do estudante. O propósito é fornecer dados para o módulo pedagógico do sistema. O módulo especialista deve ter acesso a todas as informações armazenadas.

O módulo pedagógico oferece uma metodologia para o processo de aprendizado. Questões a serem pensadas são: quando revisar, quando e como providenciar informação adicional. As entradas deste módulo são fornecidas pelo módulo estudante. As decisões pedagógicas são feitas de acordo com as necessidades individuais de cada estudante.

O módulo do domínio do conhecimento armazena a informação que o tutor está ensinando. A modelagem do conhecimento a ser disponibilizado é de grande importância para o sucesso do sistema como um todo. Critérios de desempenho também devem ser considerados. Deve-se procurar uma representação do conhecimento que esteja preparada para o crescimento incremental do domínio.

O módulo especialista deve ter acesso a todas as informações do sistema, incluindo-se o conhecimento global (domínio) e o conhecimento individual de cada estudante. A preocupação deste módulo não é a representação do conhecimento global, mas como um indivíduo representa seu próprio conhecimento através do uso de suas habilidades dentro do domínio. Usualmente, este módulo possui a forma de um sistema especialista capaz de resolver problemas em um dado domínio. Este módulo não deveria realizar apenas a comparação entre as soluções do estudante com a do tutor, mas também a comparação entre as soluções geradas pelos próprios estudantes.

O módulo de comunicação está mais voltado para o conceito de interface com estudante. A questão é definir qual será a melhor metáfora com a qual o estudante terá acesso ao sistema. A complexidade para a implementação deste módulo é bastante variável, podendo ser desde simples janelas de diálogo até linguagem natural e reconhecimento de voz. Outra questão a ser considerada é a aplicação de realidade virtual para permitir uma imersão total do estudante no sistema.

Sistemas tutores inteligentes representam uma interessante ferramenta para ambientes de aprendizagem. Entretanto, os maiores problemas associados a estes tipos de sistema são seu alto custo financeiro e o elevado tempo de desenvolvimento (Hall, 1990) (Okey et al, 1993) (Beck et al, 1998). Na tentativa de reduzir estes custos, conceitos bem conhecidos da engenharia de software como reutilização e modularidade têm sido utilizados. A questão é desenvolver sistemas ITS de forma incremental, permitindo uma evolução contínua.

2.4 Tendências Atuais da IA no Ensino

Um motivo para a redução no interesse no desenvolvimento de sistemas ITS é sua dependência na fundamentação da pesquisa em inteligência artificial e ciência cognitiva. Ciência cognitiva tem oferecido análise de tarefas de várias habilidades que permitem desenvolvedores ITS implementarem modelos cognitivos detalhados. A inteligência artificial, primeiramente na forma de sistemas especialistas e arquitetura de regras de produção, fornece meios adequados para representação e manipulação de conhecimento. Este foi o principal motivo do progresso relativamente rápido dos sistemas ITS.

Atualmente, o processo educacional está procurando novos métodos de ensino e aprendizagem. A questão é tentar reduzir a distância entre o conhecimento que um currículo tradicional acredita ser importante e o conhecimento que os estudantes acreditam ser importante (McArthur et al, 1993). Estas mudanças curriculares trouxeram a necessidade de trabalhar com problemas de mais alta ordem. As habilidades para tratar este novo enfoque não possuem ainda uma análise rigorosa da ciência cognitiva. Sendo assim, não é possível desenvolver sistemas especialistas e modelar estudantes para um sistema ITS efetivo.

Os modelos pedagógicos utilizados pelos ITS tradicionais são normalmente desprovidos de uma análise mais efetiva das interações entre aluno e sistema. Da mesma forma, os sistemas especialistas tem seus limites: não capturam todo o conhecimento do especialista humano, exigem um modelo completo do domínio, são baseados em desempenho e não para ensino e explanação. McArthur et al (1993) apresentam que o conhecimento “rico” que poderia estar contido em um sistema especialista é, muitas vezes, retirado por questões de desempenho, sem porém afetar a precisão das conclusões. Isto parece perfeito em um sistema de diagnóstico, mas sob o ponto de vista educacional, o conhecimento básico pode ser até mais importante do que regras superficiais.

2.4.1 Novos Métodos de Ensino e Aprendizado

Um método de ensino que tem sido bastante discutido nos últimos anos é baseado em investigação. Estes sistemas procuram explorar aspectos que os sistemas ITS tradicionais não consideraram. Alguns autores, como McArthur et al (1993), acreditam que estes sistemas, de modo genérico, podem ser entendidos como ambientes de ensino interativos (ILE). Os principais princípios associados a estes ambientes são:

- Construção e não instrução: o objetivo é explorar o fato de que estudantes podem aprender mais efetivamente através da construção de seu próprio conhecimento.
- O controle é do estudante e não do tutor: a questão é dar mais liberdade para o estudante controlar suas interações no processo de aprendizagem. O tutor deve atuar como um guia, e não como o único detentor do conhecimento.
- A individualização é determinada pelo estudante e não pelo tutor: diferentemente dos sistemas ITS, a personalização da informação é o resultado da interação com o ambiente. Esta responsabilidade pode estar também associada ao sistema, mas o estudante ainda terá uma boa parte do controle de sua individualização.
- O conhecimento adquirido pelo estudante é resultado de suas interações com o sistema e não com o tutor: a informação adquirida vem como uma função das escolhas e ações do estudante no ambiente de ensino e não como um discurso gerado pelo tutor.

Estes princípios apresentam uma clara mudança no enfoque de aprendizagem dos sistemas ITS para ambientes ILE. O processo deixa de ser centrado no tutor e passa a ser centrado no estudante. Sendo assim, torna-se necessária uma nova gama de ferramentas computacionais. Estas ferramentas incluem, frequentemente, vídeo interativo ou outras representações gráficas, e permitem aos estudantes investigar e aprender tópicos de forma livre, sem estarem presos a algum tipo de controle externo.

2.4.2 Aplicação de Agentes Inteligentes

Agentes inteligentes (Bradshaw, 1997) estão sendo utilizados para implementar diversos sistemas ITS (Boy, 1997), uma vez que cada componente do ambiente pode ser implementado como um agente independente, possuindo facilidades de interação com os outros agentes. A abordagem por agentes possibilita o desenvolvimento de diferentes raciocínios e a integração de várias ações para alcançar um determinado objetivo (Marietto, 1997). Além disso, a utilização de agentes inteligentes é uma boa opção para reduzir o custo destes sistemas, pois eles favorecem a modularização e a evolução.

Existem diversos exemplos na literatura sobre a utilização de agentes inteligentes em sistemas educacionais. Segundo Shoham (1993), uma sociedade de agentes para aprender e ensinar pode ser a solução para a construção de ambientes de ensino e aprendizagem, se os agentes trabalham de uma maneira concorrente e autônoma para alcançar seus objetivos. Os agentes em um ambiente de ensino/aprendizagem são considerados autônomos porque: as atividades dos agentes individuais não requerem constante supervisão externa, e não há autoridade central projetada para controlar todas as interações desempenhadas entre os agentes. O trabalho de (D'amico et al, 1997) apresenta uma arquitetura para um ambiente de ensino/aprendizagem, que utiliza agentes para realizar as tarefas necessárias deste ambiente.

As características de comunicação oferecidas pelos agentes também podem ser exploradas para ambientes distribuídos, permitindo a criação de sistemas com facilidades de colaboração. O projeto GRACILE (Ayala e Yano, 1995, 1996) discute a utilização de agentes para ambientes de ensino colaborativo em uma rede de computadores. O objetivo deste projeto foi criar um espaço virtual para que os estudantes pudessem construir sentenças da língua japonesa com a ajuda de agentes consultores. São definidos dois tipos de agentes: um agente de domínio, responsável pela representação de conhecimento e assistência pessoal, e um agente mediador, responsável pelo transporte da informação.

Agentes específicos podem ainda ser construídos para oferecer serviços adicionais a serem utilizados no processo educacional. Um bom exemplo seria oferecer serviços

automatizados de busca de material de referência, filtros inteligentes para correio eletrônico e salas de discussão.

2.4.3 Expandindo Métodos para Ensino e Aprendizado

Diversas pesquisas feitas atualmente no campo da IA podem ser utilizadas para definir e implementar uma grande variedade de métodos para ensino e aprendizado. Como visto anteriormente, o processo de investigação abre caminho para identificar a existência de várias formas de “aprender”. Esta seção procura apresentar alguns princípios que podem contribuir para a educação:

- Aprendizado baseado em casos (Schank, 1991): os estudantes adquirem novos conhecimentos a partir da exploração de situações em uma grande biblioteca de experiências passadas. O propósito é tentar aplicar soluções já testadas no problema a ser resolvido. O enfoque é fazer com que os alunos não sejam meros aplicadores de regras pré-estabelecidas, mas buscar analogias, aplicá-las e tentar explicar suas próprias regras de decisão. A seção 4.7 aborda os principais aspectos pedagógicos que podem ser explorados através do ensino baseado em casos.
- Aprendizado baseado em simulação (Schank, 1991): este princípio envolve a criação de modelos dinâmicos e simplificados do mundo real. Estes modelos permitem a exploração de situações difíceis, muito custosas ou até mesmo impossíveis de acontecerem. A simulação permite ao estudante desenvolver hipóteses, testá-las e analisar seus resultados para refinar conceitos.
- Aprendizado por necessidade: Fischer (1991) argumenta que este método situa o aprendizado num contexto de trabalho ao invés de colocá-lo numa fase separada, fazendo com que o aprendizado seja relevante à tarefa em execução. Este método é também conhecido como aprendizado *just-in-time*.
- Colaboração: tecnologias baseadas em computador para colaboração estão fornecendo novos métodos cooperativos para trabalho e aprendizado. Aplicações da IA em educação tipicamente suportam processos de aprendizado individual, não apenas porque são processos efetivos, mas porque muita pesquisa no

aprendizado em IA e ciência cognitiva está focada na cognição individual. Entretanto, a evolução das redes de computadores permite que novos métodos baseados na interação de grupos sejam possíveis.

Estes princípios demonstram que a aplicação de IA no ensino está apenas em seu estágio inicial, podendo ser explorada para uma contínua busca por novos métodos de ensino e aprendizagem.

2.5 Sistemas de Ensino Colaborativo

O aprendizado colaborativo pode ser entendido como grupos de estudantes trabalhando de forma interativa para resolver problemas. Segundo Katz (1995), estes ambientes têm sido vistos como benéficos, tanto em aspectos cognitivos como em aspectos sociais. Nesta situação, o foco não está mais na interação entre professor e estudante, mas como os estudantes podem interagir entre si e como eles podem ensinar uns aos outros sem a necessidade de um professor. Usualmente, os estudantes trabalham em conjunto através da utilização de um sistema ITS em uma rede de computadores. Entretanto, Katz (1995) ressalta que um sistema colaborativo já poderia ser alcançado apenas com a mudança na utilização dos sistemas tutores desenvolvidos para uso individual. O simples uso de um ITS local por pares de estudantes já estaria explorando os aspectos de colaboração para a solução de problemas.

No contexto deste trabalho, sistemas de ensino colaborativo apoiados por computador (CSCL – *Computer-Supported Collaborative Learning*) são ambientes desenvolvidos para permitir que vários estudantes trabalhem cooperativamente através de uma rede de computadores.

Um importante aspecto de ambientes colaborativos é que em situações de grupo, nem todos os estudantes possuem as mesmas habilidades. Ayala e Yano (1995) acreditam que a chave está em se encontrar as condições ideais que permitirão uma colaboração efetiva entre os estudantes. Por colaboração efetiva deve-se entender aquela situação onde o estudante pode aprender a partir de outros estudantes, enquanto ele realiza uma tarefa que necessita a aplicação de elementos de conhecimento internalizadas por outros estudantes e também passíveis de serem internalizadas por ele.

McArthur et al (1993) enfatiza que grupos heterogêneos podem gerar dois tipos de problema: associação de crédito e sobre como avançar no currículo. O primeiro problema trata da questão de associar o crédito de uma resposta para todos os membros do grupo ou para o estudante individual que inicialmente encontrou a resposta. O segundo problema considera as decisões pedagógicas de como avançar o grupo através do currículo. Se um estudante deveria ditar o passo de aprendizado de todo o grupo, então a questão é definir qual dos estudantes é o escolhido.

Entretanto, sistemas colaborativos não precisam ser necessariamente ambientes altamente síncronos, onde todos os estudantes envolvidos trabalham em um mesmo material simultaneamente. Foi discutido anteriormente que a individualização é uma característica importante, pois o próprio aluno pode definir a melhor forma para aprender. Sendo assim, podem ser definidos ambientes que oferecem ferramentas de colaboração assíncrona, desligadas da necessidade que o grupo todo esteja em um mesmo problema.

2.6 Colaboração Através da Internet

Um dos principais aspectos que caracterizam os ambientes de ensino colaborativo é o fato de serem suportados por redes de computadores. Como já discutido anteriormente, o ensino à distância não é um tópico novo e pode ser alcançado através de um grande número de possibilidades. Entretanto, a situação atual da Internet e das Intranets oferecem um ambiente distribuído perfeito para servir de suporte aos sistemas colaborativos para o ensino à distância.

Oliver et al (1997) apresentam que a Internet oferece a oportunidade para o desenvolvimento de ambientes de aprendizado que conectam estudantes de forma individual em comunidades virtuais compartilhando uma meta de aprendizado comum. Estes ambientes habilitam os estudantes a compartilhar recursos e materiais que vem a ser o produto do aprendizado.

Um importante ganho que pode ser obtido com a Internet está no fato dela ser inerentemente assíncrona. Neste contexto, existem as redes de aprendizado assíncronas (ALN – *Asynchronous Learning Networks*)(Bourne, 1998), onde os estudantes podem

estudar a qualquer momento, a qualquer hora e em qualquer lugar. Segundo Bourne (1998), ambientes ALN são usualmente implementadas através de *sites* na Internet e empregam conferências eletrônicas e meios de comunicação entre os participantes.

A definição de ambientes de ensino colaborativo assíncronos exige que algumas características mínimas sejam estabelecidas:

- Um componente ou ferramenta eficaz para a avaliação do progresso do estudante.
- Um ambiente efetivo, não sujeito a problemas, que permita uma boa interação entre os participantes.
- A implementação de componentes motivadores e desafiadores que procurem manter a atenção contínua do estudante.

Existem diversas propostas para o ensino através da Internet. Porter (1997) levanta uma grande variedade de ambientes que procuram criar salas de aula virtuais utilizando a Internet. Entretanto, a grande maioria não possui as características dos sistemas tutores anteriormente discutidos. Usualmente, existe um conjunto de recursos para troca de mensagens, navegação não linear no material didático, salas de discussão e meios para os alunos apresentarem suas idéias e sugestões. Apesar destes recursos serem importantes para o processo ensino/aprendizagem, não há nenhuma orientação inteligente sobre o conteúdo. Esta orientação é dada, muitas vezes, por professores ou tutores humanos em horários específicos ou através de requisições explícitas feitas pelos alunos.

2.7 Conclusão

Este capítulo procurou apresentar alguns aspectos da aplicação da Inteligência Artificial no ensino. O objetivo foi mostrar como a evolução das técnicas de IA e da ciência cognitiva influenciam diretamente na construção de sistemas de ensino apoiados por computador. Foram discutidos também aspectos que devem ser considerados para desenvolver ambientes colaborativos através da Internet.

Muitos pesquisadores têm desenvolvido software instrucional para uso colaborativo. Entretanto, poucos sistemas estão sendo construídos de forma que os recursos de ajuda

disponíveis aos estudantes sejam baseados em pesquisa empírica, onde os estudantes realizam tarefas com sucesso ou não (situação de falha) durante o processo de colaboração. Ambientes colaborativos podem aumentar a efetividade de interações de aprendizado em grupo, assumindo que estes sistemas oferecem ajuda a medida que os estudantes trabalham sobre problemas e criticam soluções de outros estudantes.

Os desenvolvedores necessitam conhecer mais sobre os tipos de conhecimento que os estudantes estão tipicamente habilitados a oferecer durante atividades de solução de problemas, e que tipos de ajuda eles podem ter de estudantes mais experientes ou professores durante os vários estágios de seu desenvolvimento instrucional. Também precisam conhecer como mentores humanos fornecem tal ajuda, para que sejam desenvolvidos novos modelos computacionais de orientação durante as interações colaborativas.

Foi verificado um grande interesse pelo desenvolvimento de sistemas de ensino colaborativo através de ambientes distribuídos. Neste contexto, a aplicação de agentes inteligentes foi discutida como uma abordagem interessante para permitir a distribuição dos sistemas de ensino em redes de computadores.

Muitos dos sistemas educacionais atuais conseguem atender apenas algumas das características desejáveis em um sistema educacional. Ambientes como o proposto em (Oliver et al, 1997) oferecem a possibilidade de colaboração de estudantes e tutores através da Internet, mas não implementam mecanismos de orientação mais elaborados. Por outro lado, existe uma grande quantidade de sistemas ITS disponíveis para diversos domínios de conhecimento. Entretanto, estes sistemas são usualmente desenvolvidos para exploração individual, não possuindo características de colaboração.

Um ponto importante é a necessidade de desenvolver novas ferramentas que possam ser integradas para alcançar diversos princípios pedagógicos. Estas ferramentas deveriam, além de permitir a cooperação dos estudantes durante a resolução de problemas, oferecer orientação e informações adicionais. Além disso, é importante que uma ferramenta voltada ao ensino permita o aprendizado também de forma assíncrona, onde o estudante possa criar sua própria programação, mas ainda permitindo a colaboração. Serviços adicionais como busca de informação automatizada, salas de discussão e correio eletrônico orientado também deveriam ser incorporados. O objetivo é fazer com

que o estudante faça uma imersão durante o tempo em que estiver utilizando a ferramenta.

Este trabalho procura contribuir com uma ferramenta baseada em agentes inteligentes que oferece facilidades de resolução de problemas através do trabalho cooperativo de seus usuários. A ferramenta foi desenvolvida para ser executada através da Internet. Sua aplicação como um importante instrumento para o ensino à distância é discutida. Procura-se mostrar como atender os principais pontos pedagógicos discutidos neste capítulo. A principal abordagem utilizada para explorar o ensino é baseada na recuperação de experiências passadas para encontrar novas soluções.

3 Teoria de Agentes

3.1 Introdução

O termo **agente** é utilizado na literatura computacional para determinar diversos tipos de programas. Estes programas não precisam, necessariamente, apresentar um comportamento “inteligente”. Obviamente, torna-se difícil estabelecer o que seria um comportamento inteligente, uma vez que o termo “inteligência” ainda é causa de controvérsia entre os estudiosos da Inteligência Artificial (IA). Discussões sobre este problema podem ser encontradas em (Wooldridge & Jennings, 1994), em seu estudo sobre agentes inteligentes, e em uma entrevista com Marvin Minsky (1994). A questão sobre estabelecer um conceito para agentes não é um problema apenas entre as diversas áreas da Computação. Segundo Franklin & Graesser (1996), mesmo entre os pesquisadores envolvidos com trabalhos referentes a agentes inteligentes, existem diversas definições sobre o tema.

Como apresentado por Hendler (1996), pode-se afirmar que os pesquisadores em agentes inteligentes procuram, de forma geral, desenvolver programas sofisticados que podem ser úteis em ambientes de importância para os humanos. Estes ambientes podem variar desde locais físicos de difícil acesso ou perigosos para os humanos até domínios virtuais, como as redes de computadores.

Trabalhos mais recentes (Spector, 1997) enfatizam que um agente pode ser qualquer sistema autônomo que percebe e age para alcançar um estreito conjunto de metas dentro de um específico ambiente virtual ou real. Os sistemas tradicionais de IA têm sido projetados para operar sob o controle do usuário e sobre um olho cuidadoso, enquanto agentes são lançados no mundo para agir autonomamente, geralmente sob suas próprias crenças. Spector (1997) ainda acredita que uma característica positiva dos agentes é sua facilidade de implementação, uma vez que são programas para atuar sob sistemas autônomos e limitados. Sendo assim, seria possível gerar programas agentes automaticamente (programação evolutiva-genética). Nesta abordagem, os agentes

seriam o resultado de uma constante evolução, onde aqueles mais especializados e com melhor desempenho seriam os sobreviventes para uma nova geração de agentes.

Pattie Maes (1996) aborda também a utilização da evolução artificial. Entretanto, ela não acredita que a construção de agentes seja tão simples, uma vez que agentes diferem de programas tradicionais principalmente pela capacidade de serem entidades independentes que devem aprender da experiência e responder a situações não previstas com um grande número de métodos diferentes.

Knapik e Johnson (1998) acreditam que agentes se tornarão cada vez mais permanentes nos sistemas computacionais e de comunicação. O motivo principal desta afirmação é a versatilidade dos agentes. Eles podem oferecer serviços como introduzir inteligência às interfaces atuais, mapear informação em conhecimento, realizar comércio eletrônico, além de atuar como procuradores para os usuários nas mais diversas situações.

Para demonstrar a grande abrangência de atuação dos agentes inteligentes pode-se verificar algumas possíveis aplicações:

- **Assistentes pessoais:** Sistemas completos e independentes que planejam, organizam e otimizam tarefas de produtividade pessoais. Aplicações possíveis incluem filtragem de correio eletrônico e sistemas de classificação, programadores automáticos de reuniões, além de diversos filtros de informação.
- **Sistema para atender visitantes:** O atendimento e programação de atividades, reuniões e eventos com a participação dos visitantes no local são organizados e coordenados através da operação de agentes locais com os assistentes pessoais dos visitantes.
- **Exploração de informação (*Data Mining*):** Agentes específicos em informação fornecem um contexto para busca de dados em grandes bancos de dados ou outras fontes de informação, que em cooperação com agentes pessoais, selecionarão a informação útil. Este campo é um dos que estão em mais rápido desenvolvimento no momento, pela explosão da necessidade de buscar informações através de redes de computadores.
- **Gerenciamento de redes:** Agentes colaborativos coletam e trocam informação local sobre as estatísticas da rede para alcançar automação e otimização das decisões nas tarefas de administração de rede tais como roteamento, acesso,

provisão de serviços, monitoramento e avaliação estatística, dentro de uma visão global.

- **Controle de Tráfego Aéreo/Terrestre:** Um sistema multi-agente pode ser instalado em locais de controle de veículos em trânsito para ajudar a resolver decisões de roteamento e programação.
- **Manufatura e Programação da Produção:** Uma coleção de agentes tentam negociar, a partir de um conjunto determinado de metas e restrições, qual a melhor sequência de eventos dentro do chão de fábrica para maximizar a produtividade.

3.2 Histórico

O conceito de decompor sistemas complexos em diferentes entidades na tentativa de obter maior eficiência não é uma abordagem recente em computação. Durante muitos anos, pesquisadores em Inteligência Artificial buscaram a construção de agentes isolados com alguma conduta inteligente. Segundo Kay (1984), a idéia de agentes foi originada em meados da década de 50 com John McCarthy e Oliver Selfridge. Eles tinham em mente um sistema onde dada uma meta, uma sequência de ações seria executada na tentativa de satisfazer aquela meta. Este sistema também poderia solicitar ajuda quando necessário. Um agente seria um robô virtual (*soft robot*) vivendo e realizando seus negócios dentro de um universo computacional.

Os primeiros agentes trabalhavam na resolução de problemas usando heurísticas ou métodos baseados em conhecimento. Alguns destes agentes poderiam ainda planejar, aprender e modificar suas percepções sobre o ambiente no qual estavam inseridos. Estas características ainda estão presentes em um grande número de aplicações da IA.

Na década de 70, Hewitt (1977) propôs um agente com o conceito de um objeto auto-suficiente, interativo e com execução concorrente, o qual ele denominava **ator**. Este objeto teria alguns estados internos encapsulados e responderia a mensagens de objetos similares.

Entretanto, o crescente interesse por paralelismo e distribuição na IA marcou a última década. De acordo com Bond (1988), o desenvolvimento de poderosos computadores, a

proliferação de redes de computadores e a consideração sobre o uso de grupos de indivíduos para resolver problemas fizeram com que a Inteligência Artificial Distribuída (IAD) se tornasse uma área emergente.

Nwana (1996) também enfatiza o crescimento da IAD e separa as pesquisas realizadas na área em duas linhas distintas: a primeira do período de 1977 até os dias atuais, e a segunda partindo de 1990. A primeira linha trabalha com agentes ativos, estando mais relacionados a tipo de agente deliberativo com modelos simbólicos internos. Um agente deliberativo é aquele que possui uma representação explícita, um modelo simbólico do mundo, e cujas decisões (ações a executar) são realizadas através de um raciocínio simbólico (Wooldridge e Jennings, 1994). A segunda linha de pesquisa, em rápido desenvolvimento, enfatiza a diversificação das tipologias ou classes de agentes, onde os agentes apresentam uma evolução em sua inteligência e autonomia. Bradshaw (1997) afirma que esta linha demonstra a evolução da deliberação para a execução, do raciocínio para a ação remota.

Chaib-draa (1995) considera que a motivação para a IAD pode ser caracterizada pelos seguintes tópicos:

- Necessidade de manipular conhecimento distribuído em aplicações geograficamente dispersas.
- Ampliação da cooperação homem-máquina através da distribuição de homens e máquinas.
- Inovações na representação de conhecimento e resolução de problemas, aproximando-se mais da realidade.
- Descoberta de um novo foco sobre as ciências cognitivas e inteligência artificial.

A IAD pode ser vista como sendo um subcampo da Inteligência Artificial voltado para a construção de agentes inteligentes. A IAD está diretamente ligada a agentes inteligentes, uma vez que estes podem ser vistos como entidades dentro de um sistema inteligente distribuído. Estas entidades são autônomas e podem cooperar entre si através de algum mecanismo de natureza inteligente. Segundo Chaib-draa (1995), os agentes precisam

tomar decisões que permitam alcançar os seus objetivos em um mundo povoado por outros agentes inteligentes e com seus próprios objetivos.

Oliveira (1996) define IAD como sendo o estudo computacional da conduta inteligente, resultando na interação de diversas entidades autônomas. Estas entidades são usualmente chamadas de agentes, e o sistema como um todo é usualmente chamado de sociedade. Os conceitos de interação e conduta social também estão presentes no trabalho de Sichman et al (1992). Sichman considera que estes aspectos podem ser utilizados para a construção de uma inteligência coletiva. Este item em particular diferencia a IAD da IA tradicional, a qual trabalha com representação do conhecimento e métodos de inferência voltados para a construção de uma inteligência individual.

A Inteligência Artificial Distribuída pode ser subdividida em duas grandes áreas¹:

- Sistemas Multi-Agentes (MAS - *Multi-Agent Systems*)
- Resolução Distribuída de Problemas (DPS - *Distributed Problem Solving*)

É importante ressaltar que o termo MAS tem sido utilizado, atualmente, para definir quaisquer sistemas onde múltiplos agentes interagem (Durfée & Rosenschein, 1994). Entretanto, a divisão apresentada procura mostrar os diferentes tipos de problemas, além de tentar evitar o mau uso do termo. De acordo com Scherer & Schlageter (1995), o principal ponto de pesquisa em MAS está focado na coordenação. Estes autores continuam afirmando que as abordagens MAS atuais possuem um aspecto inerentemente estático. MAS são projetados para seguir uma filosofia *top-down*, onde, normalmente, não é permitido a adição ou remoção dinâmica de agentes. Além disso, ambientes MAS têm usado um forte acoplamento, onde os agentes não têm capacidade de decidir por si próprios se devem ou não cooperar.

Tanto em problemas MAS quanto em problemas DPS, existem diversos agentes compartilhando um mesmo ambiente. A principal diferença está na forma de interação entre os agentes. Em um modelo DPS, os agentes buscam a cooperação para alcançar suas metas (por este motivo, problemas deste tipo são também conhecidos por

¹ Embora a maioria dos autores concorde com esta organização, alguns autores como Nwana (1996) apresentam ainda uma terceira grande área, denominada Inteligência Artificial Paralela.

problemas DPS cooperativos). Neste caso, os agentes lidam com problemas que estão além de suas capacidades individuais e apenas a cooperação mútua permitirá a resolução do problema global. Em um modelo MAS, os agentes competem por recursos e precisam atuar coletivamente para identificar e resolver conflitos. Ao mesmo tempo, os agentes procuram se beneficiar com as ações tomadas por outros agentes.

Segundo Durfee & Rosenschein (1994), sistemas DPS procuram trabalhar na base da confiança, sinceridade, divisão de tarefas e outras propriedades que são muito difíceis de serem alcançadas em uma coleção de indivíduos. Sendo assim, os sistemas MAS basearam-se na teoria dos jogos para demonstrar que um agente precisa ser racional², ou seja, todas as suas ações devem ser no sentido de maximizar o seu próprio custo/benefício.

Entretanto, o trabalho de Durfee & Rosenschein não procurou separar totalmente sistemas DPS de sistemas MAS, mas apresentou três visões de relacionamento entre eles³. Em uma primeira proposta, sistemas DPS podem ser vistos como sendo um subconjunto dos sistemas MAS. Nesta situação, um sistema MAS assegura algumas propriedades como metas comuns e projeto centralizado. Na segunda proposta, os sistemas MAS fornecem a base para os sistemas DPS. Sendo assim, um sistema MAS procura trabalhar com a individualidade e o interesse próprio de cada agente para definir como serão feitas as interações com outros agentes. O sistema DPS procura então, verificar como estas interações podem ser exploradas de forma a resolver algum problema mais global. Na terceira proposta, os sistemas MAS são vistos como sendo complementares aos sistemas DPS. Neste caso, um sistema MAS procura responder de que forma algumas propriedades coletivas podem ser percebidas em um ambiente particular. Enquanto, sistemas DPS procuram responder como uma coleção particular de agentes pode obter algum desempenho eficiente se as propriedades do ambiente são dinâmicas e não controláveis.

² O termo racional aqui possui um sentido menos abrangente do que aquele apresentado por Russell & Norvig (1995), onde a racionalidade está ligada à habilidade de um agente agir de acordo com as suas crenças, na busca de suas metas.

³ Durfee & Rosenschein estabelecem que as formas de relacionamento não devem ser vistas como sendo mutuamente exclusivas, mas como tendo pontos de intersecção.

3.3 Definições de Agentes

Como enfatizado anteriormente, existem diversas definições para agente. Normalmente, estas definições estão associadas a diferentes pontos de vista e dependem muito da funcionalidade fornecida pelo agente em questão. Entretanto, a visão de vários conceitos auxilia no processo de entendimento de aspectos importantes que envolvem o termo agente. Este trabalho procura abordar apenas agentes de software, embora muitas das características que serão discutidas poderem estar presentes em agentes de hardware.

Antes de abordar conceitos mais técnicos, este trabalho considera importante verificar qual o significado da palavra agente na língua portuguesa para se ter uma idéia inicial sobre o tema. O dicionário interpreta agente como:

agente n.

1. Alguém que atua
2. alguém atuando ou fazendo negócios por outro
3. procurador, delegado.

... (Merriam-Webster, 1996).

Apesar das definições supracitadas atenderem algumas características, agentes inteligentes possuem um significado bem mais amplo do ponto de vista da comunidade de pesquisadores em Inteligência Artificial. Algumas destas abordagens serão vistas nesta seção.

Wooldridge e Jennings (1994) afirmam que agentes são sistemas que apresentam um comportamento determinado por um processo de raciocínio baseado na representação de suas atitudes, tais como crenças, compromimentos e desejos. Eles acreditam que um sistema pode ser visto como um agente se ele possuir as seguintes propriedades:

- **Autonomia:** o agente deve poder funcionar sem intervenção humana, baseando suas ações em seu conhecimento armazenado sobre o ambiente.
- **Habilidade Social:** o agente interage com outros agentes através de uma linguagem comum.
- **Reatividade:** o agente deve ser capaz de perceber mudanças em seu ambiente e atuar de acordo com estas mudanças.

- **Pró-Atividade:** o agente não deve apenas atuar por percepção, mas deve procurar alcançar uma meta apresentando iniciativa.

Seguindo a mesma linha de raciocínio da definição anterior, Genesereth and Ketchpel (1994) descrevem agentes como componentes de software que se comunicam com seus pares através da troca de mensagens em uma expressiva linguagem de comunicação de agentes.

Uma definição mais voltada para o uso de agentes como assistentes pessoais é proposta por Maes (1994). Ela define agentes como sendo componentes de software que atuam autonomamente de forma a atender os interesses do usuário.

Russell e Norvig (1995) apresentaram uma nova proposta de entender Inteligência Artificial. Em seu trabalho, eles estabelecem que inteligência está altamente ligada com ações racionais. Agir racionalmente significa agir de forma a alcançar as metas definidas por alguém, dadas as crenças deste alguém. Neste sentido, ele definem genericamente um agente como sendo algo com a capacidade de atuar sobre um ambiente através de sua percepção sobre este ambiente.

Franklin e Graesser (1996) procuraram não apenas definir um agente, mas estabeleceram uma taxonomia baseada na principal noção de autonomia. Eles consideram que um agente autônomo é um sistema que faz parte de um ambiente, onde ele percebe e atua para atender suas próprias metas.

Nissen (1995) utiliza uma definição mais informal. Ele define um agente como sendo alguém ou alguma coisa que atua como um procurador com o propósito específico de realizar ações que podem ser entendidas como benéficas à parte representada.

Agentes podem também ser definidos operacionalmente em termos de domínios no qual fornecem seus serviços (Knapik e Johnson, 1998), incluindo busca por informação, filtragem de dados, ajuda sensível ao contexto, assistência em tempo real, além da execução de tarefas conforme as necessidades de um usuário e outros. Neste contexto, uma abordagem possível para definir agentes pode ser através de sua organização, seguindo algum tipo de classificação.

Uma definição mais específica e que deve ser aceitável pela maioria dos pesquisadores é proposta por (Shoham, 1997). Ele define agente como sendo uma entidade de software funcionando continuamente e de forma autônoma em um ambiente particular, freqüentemente habitado por outros agentes e processos.

Como visto, existem diversas propostas para definir o que é um agente. Entretanto, pode-se observar claramente que todas possuem características em comum. Dentre estas características deve-se enfatizar propriedades como autonomia, capacidade de responder a determinadas situações, facilidades para comunicação e capacidade para aprender como alcançar seus objetivos. Por serem muito importantes no entendimento de agentes, estas propriedades serão exploradas na próxima seção.

3.4 Atributos dos Agentes

Foi discutido anteriormente que um agente pode ser definido a partir de suas características básicas. O conjunto destas características é ainda utilizado como uma forma para agrupar os agentes em classes ou tipologias. Um agente não precisa possuir todas estas características ou atributos, embora suas capacidades estejam diretamente associadas a presença delas.

Esta seção não procura apresentar todos os atributos existentes, mas apenas os mais relevantes. É importante ainda ressaltar que os atributos discutidos não estão em ordem de importância. A escolha de quais atributos devem estar presentes em um agente depende da funcionalidade que o projetista pretende dar ao seu agente.

3.4.1 Autonomia

Segundo Nwana (1996), autonomia refere-se ao princípio de que os agentes podem agir baseados em seus próprios princípios, sem a necessidade de serem guiados por humanos. Os agentes possuem estados e metas internos, agindo de maneira a atingir estas metas em favor de seus usuários. O elemento chave da autonomia é a proatividade, que é a sua habilidade de tomar iniciativas, sem a necessidade de agir em virtude de uma mudança de seu ambiente (Wooldridge e Jennings, 1994).

Nissen (1995) relaciona o controle somente ao usuário final quando estabelece que um agente inteligente deve possuir a habilidade de praticar ações para desenvolver tarefas ou alcançar objetivos, sem necessitar da interferência do usuário final.

De acordo com Foner (1993), os agentes detentores de um alto grau de autonomia podem manter suas agendas independentes daquelas de seus usuários. Para tanto, os agentes apresentam 3 aspectos básicos: ações periódicas, execução espontânea e iniciativa. Estes requisitos habilitam o agente autônomo a efetivar ações preemptivas e independentes que poderão eventualmente beneficiar o usuário.

No contexto da Internet, Nissen (1995) afirma que autonomia é a habilidade de operar, mesmo quando o usuário final está desconectado da rede. Neste caso, esta propriedade normalmente está associada a outra: mobilidade.

3.4.2 Mobilidade

A capacidade de poder se mover através de uma rede de computadores parece ser interessante para agentes que auxiliam seus usuários na busca de informações, principalmente dentro da Internet. Entretanto, este atributo pode causar sérios problemas de sobrecarga na rede, uma vez que eles trafegam entre as máquinas conectadas. Knapik e Johnson (1998) apresentam outro problema que está relacionado com segurança: um agente móvel pode conter problemas de código ou até mesmo estar transportando um vírus de computador. Sendo assim, a implantação de agentes móveis deve ser acompanhada de processos de autorização, além da garantia de que a memória e os recursos da máquina estarão protegidos.

3.4.3 Cooperação

Cooperação pode ser entendida como a capacidade que os agentes tem de trabalharem em conjunto de forma a concluírem tarefas de interesse comum. Nwana (1996) acredita que a cooperação entre agentes é fundamental, sendo a razão principal para a existência de um ambiente multi-agente. Para permitir esta cooperação, o agente deve ser dotado de uma certa habilidade social, capacitando-o a interagir com outros agentes e possivelmente humanos através de alguma linguagem de comunicação (Wooldridge e Jennings, 1994).

3.4.4 Comunicabilidade

Quando existe mais de um agente envolvido, há uma necessidade óbvia por um modelo de comunicação. Entretanto, o conceito de comunicabilidade não estabelece apenas a troca de informações entre agentes. Franklin e Graesser (1996) afirmam que agentes podem se comunicar com outras entidades além de agentes, incluindo-se humanos e o seu ambiente.

Por ser um atributo importante no contexto deste trabalho, a seção 2.6 discute a comunicação entre agentes com maior detalhe.

3.4.5 Aprendizagem

Um dos atributos que mais caracterizam agentes inteligentes é a capacidade de aprender. Uma real autonomia só pode estar presente quando um agente possui a habilidade de avaliar as variações de seu ambiente externo e escolher qual a ação mais correta. Entretanto, mesmo quando um agente não reconhece nenhuma ação a ser executada, é esperado que ele procure encontrar uma saída. A questão não é acertar sempre, mas aprender continuamente por experiência, seja através de sucessos ou de fracassos.

O aprendizado pode ser também um processo iterativo. Nestes casos, o treinador pode fornecer ao agente o conhecimento através de uma sequência de instruções ou informá-lo apenas quando o agente não possuir o conhecimento necessário (Lemon et al, 1996).

3.4.6 Reatividade

Reatividade é a habilidade que um agente tem de reagir a mudanças no seu ambiente. Para tal, o agente deve ser capaz de perceber seu ambiente e atuar sobre ele. Este atributo está presente em praticamente todas as definições de agente (Wooldridge e Jennings, 1994), (Russell e Norvig, 1995), (Franklin e Graesser, 1996).

3.4.7 Habilidade Social

Este atributo está diretamente associado com a característica de comunicabilidade, uma vez que representa a habilidade de interagir com outros agentes. Genesereth e Ketchpel

(1994) indicam a necessidade de uma linguagem comum para a comunicação de agentes.

3.4.8 Pró-atividade

Este atributo pode ser também denominado iniciativa, uma vez que representa um comportamento independente. As ações são dirigidas pelo objetivo e não simplesmente por mudanças no seu ambiente. O agente que implementa este atributo possui maior flexibilidade, pois é capaz de resolver problemas causados por situações inesperadas.

3.5 Tipologia de Agentes

A grande quantidade de atributos discutidos na seção anterior ajuda a perceber que seria muito difícil implementar um agente que incorporasse todos aqueles atributos. Até porque as características de um agente são dependentes do tipo de aplicação a que ele se propõe.

A análise dos atributos que estão presentes nos agentes tem sido utilizada pelos pesquisadores para organizar os agentes em tipologias. Uma tipologia é uma classificação por tipos de agentes que possuem atributos em comum. Alguns dos principais atributos utilizados na formação de uma tipologia foram discutidos na seção anterior (3.4).

Nwana (1996) propõe uma tipologia de agentes que identifica diferentes dimensões de classificação. Ele acredita que agentes podem ser classificados de acordo com:

- Sua mobilidade: estático ou móvel. No caso de ser móvel, o agente pode ainda estar residente na máquina do cliente ou temporariamente no servidor.
- A presença ou não de um modelo de raciocínio simbólico, ou seja, um agente pode ser deliberativo ou reativo.
- A presença dos atributos primários autonomia, cooperação e aprendizado. Combinando estas três características, quatro tipos de agentes podem ser derivados: agentes colaborativos, agentes colaborativos com capacidade de aprendizado, agentes de interface e agentes verdadeiramente inteligentes (fig. 2). É importante notar que os limites desta classificação não devem ser

interpretados como linhas bem definidas. Na verdade, o fato de agentes cooperativos terem mais ênfase em cooperação e autonomia que agentes com capacidade de aprendizado não exclui a possibilidades daqueles desenvolverem características de aprendizado.

- A função principal assumida pelo agente, como os agentes de informação.
- Suas características híbridas que combinam duas ou mais filosofias diferentes em um mesmo agente.

Após estabelecer sua tipologia, Nwana estabeleceu 7 categorias de agentes: agentes colaborativos, agentes de interface, agentes móveis, agentes de informação, agentes reativos, agentes híbridos e agentes inteligentes.

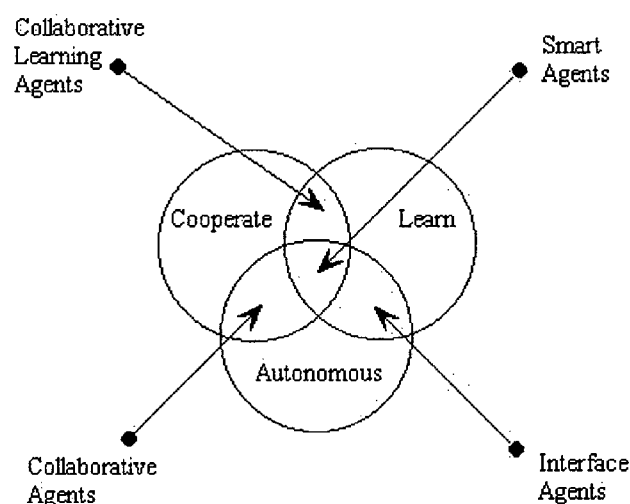


Fig. 2 – Tipologia de Agentes proposta em (Nwana, 1996).

Franklin e Graesser (1996) acreditam que um agente, por definição, deve ser um processo em contínua execução⁴ e possuir, pelo menos, os atributos de autonomia, reatividade e pró-atividade. Wooldridge e Jennings (1994) acrescentam ainda que um agente precisaria também possuir a habilidade social. Franklin e Graesser (1996), após estudar várias definições de agentes, apresentaram uma taxonomia que engloba grande parte dos trabalhos em andamento (fig. 3).

⁴ É importante notar que Franklin e Graesser consideram o fato de um agente ser um processo em contínua execução como sendo um atributo. Entretanto, este trabalho acredita ser muito óbvio para ser caracterizado de tal forma.

Caglayan e Harrison (1997) estabelecem uma taxonomia através dos conceitos de ambiente, tarefa e arquitetura. Por ambiente deve-se entender o universo onde o agente atua, como um sistema operacional em particular ou a Internet. O conceito de tarefa está relacionado com o quê um agente pode fazer. Arquitetura representa como o agente organiza internamente seu conhecimento. Utilizando estes conceitos, os autores dividiram os agentes em 3 categorias: *desktop*, Internet e Intranet. Agentes *Desktop* são basicamente agentes de interface que oferecem serviços de assistência aos seus usuários. Na categoria Internet, os agentes oferecem serviços de busca, filtragem e recuperação de informações. Os agentes móveis e de notificação também estão inclusos nesta categoria. Os agentes da categoria Intranet incluem aqueles que possuem capacidade para automatizar processos do fluxo de trabalho dentro das organizações. Outros agentes Intranet manipulam bancos de dados e alocam recursos em uma arquitetura cliente/servidor.

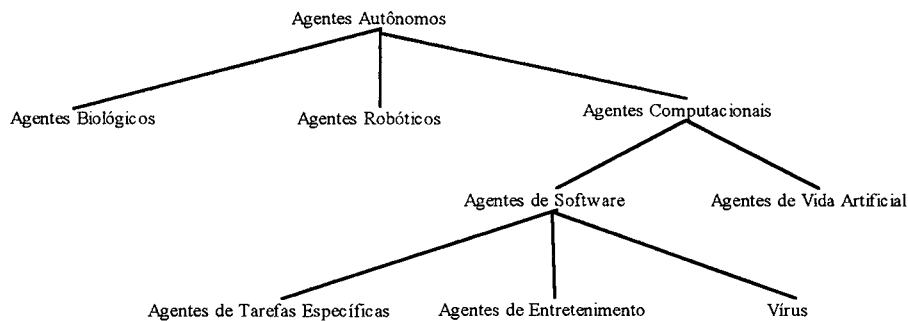


Fig. 3 – Taxonomia de Agentes proposta por Franklin e Graesser (1996).

3.5.1 Agentes Colaborativos

Em sistemas colaborativos, cada agente contribui com sua própria técnica inteligente para a solução de um problema complexo. Agentes colaborativos enfatizam autonomia e cooperação com outros agentes de forma a executar tarefas para seus donos. Neste ambiente, torna-se clara a necessidade de negociação para estabelecer acordos e comprometimentos mútuos. Apesar de aprendizado não ser a principal ênfase da operação de agentes colaborativos, eles podem demonstrar um aprendizado limitado. Para que um processo de colaboração possa acontecer, é clara a necessidade de se definir uma linguagem comum para a comunicação entre agentes.

3.5.2 Agentes de Interface

Os agentes de interface podem ser conhecidos também como sendo agentes que podem aprender ou assistentes pessoais. Eles enfatizam a autonomia e o aprendizado para executarem tarefas para seus donos. Este tipo de agente atua normalmente em *background*, analisando as ações do usuário, encontrando padrões repetitivos e automatizando estes padrões com a aprovação do usuário.

A idéia de utilizar agentes inteligentes como assistentes pessoais (Minsky, 1994) torna-se mais clara a medida que a tecnologia de agentes evolui. Segundo Fleischhauer (1996), esta nova tecnologia vem aproximar ainda mais o usuário do seu computador pessoal. Maes (1994) também idealiza estes agentes através da metáfora de um assistente pessoal que colabora com o usuário em um mesmo ambiente de trabalho. Torna-se importante verificar que a colaboração citada não precisa utilizar uma linguagem explícita para a comunicação, uma vez que ela é feita diretamente com o usuário e não com outros agentes.

Essencialmente, agentes de interface suportam e providenciam assistência, tipicamente para o usuário aprender a usar uma aplicação em particular, como um sistema operacional, por exemplo. O agente observa o usuário e monitora suas atividades na interface, aprendendo maneiras novas de executar tarefas, e sugerindo maneiras melhores de executá-las. Desta maneira, conforme Nwana (1996), o agente atua como um assistente pessoal autônomo que coopera com o usuário realizando algumas tarefas na aplicação. Os agentes de interface aprendem para oferecerem um melhor auxílio aos seus usuários. Segundo Maes (1994), os agentes de interface podem aprender a partir de quatro maneiras:

- Observando e imitando o usuário (aprendendo a partir do usuário).
- Recebendo do usuário retorno positivo e negativo (aprendendo a partir do usuário).
- Recebendo instruções explícitas do usuário (aprendendo a partir do usuário).
- Solicitando orientação para outros agentes (aprendendo a partir de parceiros).

Atualmente, há um grande número de aplicações na Internet que dizem possuir a habilidade de aprendizado. Normalmente, estes agentes estão relacionados com recomendação de serviços, tais como livros ou músicas.

Um bom exemplo de agente de interface é o Open Sesame! (Caglayan et al, 1996). Este agente utiliza uma abordagem híbrida para a aquisição de conhecimento e mecanismos de inferência que combina redes neurais e sistemas especialistas.

3.5.3 Agentes Assistentes

No contexto deste trabalho, agentes assistentes são agentes pessoais inteligentes. Apesar de possuírem semelhanças com os agentes de interface, os assistentes adicionam facilidades de ajuda, diagnóstico e orientação, além de poderem executar tarefas autonomamente, sem a intervenção humana.

Um assistente deve ser capaz de identificar situações onde o usuário pode precisar de ajuda e, então, fornecer alguma informação adicional para auxiliá-lo. Outra facilidade é a habilidade de representar seu usuário de forma autônoma, utilizando uma base de crenças criada a partir dos interesses de seu usuário. Desta forma, humanos poderão participar de um processo cooperativo envolvendo outros humanos e agentes.

Um exemplo de agente assistente é o agente de domínio do projeto GRACILE (Ayala e Yano, 1995, 1996). O contexto deste agente é um ambiente de aprendizado colaborativo para o ensino da língua japonesa. Este agente é capaz de assistir alunos através da apresentação de estruturas de linguagem e realização de análises sobre as construções produzidas por eles.

3.5.4 Agentes de Recuperação de Informação

Estes agentes são capazes de buscar informação de uma forma inteligente. É importante notar que estes agentes não devem ser confundidos com simples mecanismos de busca utilizados na Internet. O objetivo não é simplesmente encontrar informações que satisfaçam um conjunto de palavras-chave, mas espera-se que este tipo de agente possa reconhecer padrões de informação e encontrar aquelas mais relevantes. Além disso, este agente deve poder operar em modo autônomo, realizando filtragens e em alguns casos

aplicando inferências. Neste caso, o agente consegue transformar pedaços de informação em conhecimento altamente produtivo para seu usuário. Estes agentes tem ampla aplicação em organizações que possuem grande volume de informação espalhada geograficamente ou em vários bancos de dados.

3.6 Comunicação entre Agentes

Os sistemas modernos de computação freqüentemente envolvem múltiplos computadores interagindo de forma distribuída (Finin et al, 1994). Neste sentido, a habilidade de comunicação é um importante atributo que os agentes inteligentes devem possuir. Ambientes que possuem mais de um agente, praticamente exigem o intercâmbio de informações. Neste sentido, torna-se claro a necessidade de uma linguagem de comunicação comum. Apesar desta afirmação ser clara e bem aceita pela comunidade de pesquisadores, o problema é definir qual a linguagem ideal. Atualmente, não existe uma linguagem padronizada e aceita mundialmente para a representação de informações trocadas por agentes. A presença de uma linguagem comum para troca de mensagens é uma das características que diferencia um agente de um objeto na programação orientada a objetos. Uma mensagem no contexto dos agentes carrega uma semântica independente do agente, enquanto que uma mensagem no contexto dos objetos pode variar de um objeto para outro.

Finin et al (1993) estabelecem que existem diversos níveis nos quais sistemas baseados em agentes devem interagir:

- **Transporte:** como agentes enviam e recebem mensagens.
- **Linguagem:** qual o sentido de mensagens individuais.
- **Política:** como os agentes estruturam conversações.
- **Arquitetura:** como conectar sistemas em concordância com protocolos existentes.

O projeto de uma linguagem de comunicação entre agentes utiliza normalmente uma das seguintes abordagens: procedural ou declarativa. Na abordagem procedural, a comunicação acontece através de diretivas. Tanto comandos individuais quanto programas completos podem ser transmitidos e executados no lado receptor.

Linguagens baseadas em *scripts*, tais como TCL e *Telescript*, são exemplos da abordagem procedural.

Na abordagem declarativa, a comunicação ocorre através da troca de estruturas declarativas, tais como definições, asserções e outras. A abordagem declarativa pode ser encontrada no trabalho de Genesereth e Ketchpel (1994), que escolheram a Linguagem de Comunicação entre Agentes (ACL - *Agent Communication Language*) como a base de comunicação para sua plataforma de Engenharia de Software baseada em agentes. A ACL foi o resultado do grupo ARPA KSE (*Knowledge Sharing Effort*) (Neches et al, 1991). Pode-se dividir a ACL em três partes:

- **Vocabulário:** funciona como um grande dicionário de palavras apropriadas às áreas de aplicação comum. Este vocabulário compartilhado é chamado de ontologia.
- **Linguagem interna:** KIF (*Knowledge Interchange Format*) é uma versão do cálculo de predicados de primeira ordem. Possui a capacidade de codificar dados simples, restrições, regras, expressões e outras. Uma completa descrição da linguagem KIF e de como utilizá-la pode ser encontrada no trabalho de Genesereth e Fikes (1992).
- **Linguagem externa:** KQML (*Knowledge Query and Manipulation Language*) é uma camada lingüística que pode encapsular estruturas KIF. Ela fornece informação contextual para uma comunicação mais eficiente. Uma das especificações iniciais de KQML pode ser encontrada em (Finin et al, 1993).

Uma mensagem ACL é uma expressão KQML na qual os argumentos são termos ou sentenças KIF formadas por palavras no vocabulário ACL. Entretanto, é importante ressaltar que a linguagem KQML não foi idealizada para transmitir apenas um conteúdo baseado em KIF. Na verdade, KQML pode ser bem mais abrangente, permitindo até a transmissão de um conteúdo definido pelos próprios projetistas do sistema multi-agente em questão. Obviamente, a não utilização de uma linguagem padronizada e com habilidade de representação de conhecimento, como KIF, reduz a possibilidade de interação entre agentes desenvolvidos por projetistas diferentes.

Por ser importante para o contexto deste trabalho, a linguagem KQML será apresentada em mais detalhes na próxima subseção.

3.6.1 KQML

A linguagem KQML fornece uma plataforma para programas e agentes trocarem informações e conhecimento. KQML está focada nos formatos de mensagem e em protocolos de manipulação destas mensagens entre agentes em execução. Entretanto, KQML não se preocupa com o formato da informação propriamente dita. Suas expressões usualmente encapsulam estruturas de outras linguagens denominadas “linguagens de conteúdo”. Como visto, o grupo KSE utiliza a linguagem KIF como responsável por carregar o conteúdo. KQML é uma linguagem que permite programas realizarem operações sobre as bases de conhecimento de cada agente envolvido.

É importante notar que, por se tratar de uma linguagem, alguns termos apresentados neste trabalho não serão traduzidos. Uma mensagem KQML é chamada *performative*. Cada mensagem tem o objetivo implícito de realizar alguma ação específica. Como pode ser observado na especificação da linguagem (Finin et al, 1993), existe um grande número de *performatives* definidas e a maioria dos sistemas baseados em agente suportam somente um pequeno subconjunto delas. As *performatives*, ou tipos de mensagem, são palavras reservadas em KQML. Usando *performatives*, agentes podem perguntar a outros agentes por informações, dizer a outros agentes fatos, divulgar seus serviços a outros agentes e solicitar serviços de outros agentes.

KQML adota o uso de ontologias. Ontologias são um conjunto de especificações explícitas de significado, conceitos e relacionamentos aplicáveis a algum domínio específico. Desta forma, pode-se assegurar que dois agentes estejam utilizando a mesma linguagem durante o processo de comunicação. Em outras palavras, a utilização de uma ontologia permite a definição de um contexto único, eliminando-se a ambigüidade.

Mensagens KQML codificam informação em três diferentes níveis arquiteturais: conteúdo, mensagem e comunicação. Sua sintaxe está baseada na linguagem Lisp e é composta de uma ação (*performative*) e parâmetros. A ordem de posicionamento dos parâmetros não é importante. Os parâmetros são codificados como pares <palavra-chave valor>, onde a palavra-chave é precedida pelo símbolo “.”.

Um exemplo de uma mensagem KQML é apresentado na figura 4. No exemplo, um agente *user1* solicita o conceito de um objeto (em um provável curso de programação orientada a objetos) para outro agente *user2*.

A *performative* KQML utilizada é *ask-one*. O parâmetro *:content* define o nível de conteúdo da mensagem com uma estrutura da linguagem Prolog que espera retornar em *X*, o conceito de um objeto. Os parâmetros *:sender* e *:receiver* especificam a informação ao nível de comunicação. O nome da *performative*, a especificação da linguagem (*:language*) e o nome da ontologia usada (*:ontology*) fazem parte do nível de mensagem. É importante notar que a mesma mensagem poderia ser enviada com um conteúdo e linguagem diferentes, ou ainda, exatamente a mesma mensagem com uma ontologia diferente. Neste caso, o significado da mensagem seria outro, completamente diferente do mostrado no exemplo.

```
(ask-one
  :sender user1
  :content concept(object, X)
  :receiver user2
  :reply-with object-concept
  :language Prolog
  :ontology TEACH.COURSE)
```

Fig. 4 – Exemplo de uma mensagem KQML.

Uma característica interessante da linguagem KQML é que ela pode oferecer uma forma de acesso à informação, mesmo para programas que não sejam agentes. Segundo Finin et al (1993), a implementação de um agente não é necessariamente estruturada como sendo uma base de conhecimento. Sua implementação pode ser um simples sistema de banco de dados ou um programa utilizando um estrutura de dados especial encapsulada por uma interface de acesso semelhante a uma base de conhecimento. Isto permitiria que outros agentes tivessem acesso às informações disponíveis. Desta forma, cada agente poderia gerenciar uma base de conhecimento virtual (VKB – *Virtual Knowledge Base*).

3.7 Arquiteturas de Agentes

Uma arquitetura de software pode ser descrita como sendo a configuração dos componentes que constituem um sistema e das conexões que coordenam as atividades entre estes componentes (Abowd et al, 1996). Entretanto, uma arquitetura para agentes refere-se ao modo de organização dos agentes dentro de um sistema e como estão estruturados seus relacionamentos e interações.

Assim como existem diversas arquiteturas de software, o mesmo ocorre com relação as arquiteturas de agentes, as quais possuem certas características que permitem a avaliação de sua qualidade e eficácia.

Segundo Wooldridge e Jennings (1994), uma arquitetura de um agente pode ser estruturada através de uma metodologia específica para definir agentes. Sendo assim, a arquitetura abrangeria técnicas e algoritmos para suportar esta metodologia.

Conforme Knapik e Johnson (1998), a discussão sobre a qualidade de uma arquitetura de agentes torna-se subjetiva, uma vez que os detalhes desta discussão dependem de aspectos específicos da aplicação agente que se pretende desenvolver. No entanto, Mowbray (1995) procura estabelecer alguns conceitos que podem ser úteis para o desenvolvimento de uma arquitetura promissora:

- **Simplicidade:** idealizar a arquitetura e seus componentes de forma que sejam fáceis de entender, implementar e manter.
- **Funcionalidade:** selecionar uma arquitetura e ferramentas de desenvolvimento que focalizem aspectos específicos do problema a ser abordado.
- **Expansividade:** a arquitetura deve poder ser ampliada, uma vez que nem todas as necessidades futuras podem ser previstas em um primeiro momento.
- **Isolamento ou Portabilidade:** uma arquitetura, para poder ser expansiva, deve possuir uma implementação portátil, evitando-se soluções não padronizadas.

3.7.1 Classificação de Arquiteturas

Em computação, o termo arquitetura pode compreender uma faixa razoável de possibilidades, principalmente no aspecto complexidade. Arquiteturas de agentes não

são uma exceção e podem ser classificadas de acordo com as necessidades da aplicação, dos usuários, e o grau de sofisticação ou nível de inteligência dos agentes. De acordo com Knapik e Johnson (1998), a complexidade de uma arquitetura pode ser classificada em três grupos:

- **Arquitetura simples:** quando é composta por um único e simples agente.
- **Arquitetura moderada:** quando é composta por alguns agentes que realizam as mesmas tarefas, mas possuem diferentes usuários e podem residir em máquinas diferentes.
- **Arquitetura complexa:** quando é composta por diferentes tipos de agentes, cada um com certa autonomia, podendo cooperar e estar em diferentes plataformas.

Wooldridge e Jennings (1994) se baseiam na forma de construção dos agentes envolvidos para dividir as arquiteturas em três áreas:

- **Arquitetura deliberativa:** segue a abordagem clássica da Inteligência Artificial, onde os agentes atuam com pouca autonomia e possuem modelos simbólicos de seus ambientes. Em outras palavras, esta arquitetura interpreta os agentes como parte de um sistema baseado em conhecimento.
- **Arquitetura reativa:** esta abordagem procura não utilizar nenhum tipo de modelo ou raciocínio simbólico. Este tipo de arquitetura baseia-se na proposta que um agente pode desenvolver inteligência a partir de interações com seu ambiente, não necessitando de um modelo pré-estabelecido.
- **Arquitetura híbrida:** como o próprio nome sugere, este tipo de arquitetura combina características das duas abordagens anteriores.

3.7.2 Arquitetura M

Esta arquitetura, proposta por Riecken (1994), adota a premissa que um assistente pode ser implementado através de vários agentes integrados. Ela resultou na idealização de M, um programa assistente que procura reconhecer, classificar, indexar, armazenar, recuperar, explicar e apresentar informações relacionadas à interação homem-computador em um ambiente de conferência multimídia.

M é uma arquitetura complexa, envolvendo a interação e a coordenação de vários agentes direcionados a objetivos, os quais colaboram e contribuem na realização de tarefas complexas.

A arquitetura M é composta por vários tipos de componentes que providenciam capacidades espaciais, estruturais, funcionais, temporais, causais, baseadas em explanação, e raciocínio baseado em casos. Seu desenvolvimento baseou-se na decomposição de tarefas. A figura 5 ilustra como as diferentes tarefas estão associadas à diferentes tipos ou classes de agentes e outros componentes funcionais.

O componente **quadro negro** (*Blackboard*) corresponde a uma estrutura global usada pelo sistema para armazenar dados relacionados a tarefas e as entidades envolvidas nesta estrutura.

As **redes semânticas** encapsulam parte do conhecimento em nós conectados por relacionamentos. Quando um nó é ativado, ou seja, determinado conhecimento é acessado por outro agente ou componente, outros nós também podem ser acessados ou ativados, através dos relacionamentos.

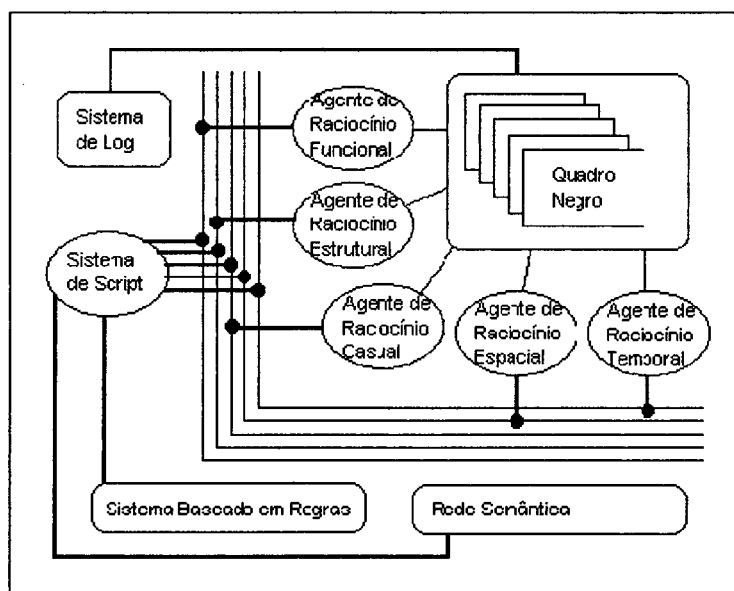


Fig. 5 – Arquitetura de Software M (Riecken, 1994).

O **sistema baseado em regras** utiliza a abordagem clássica para a definição da base de conhecimento que consiste de fatos e regras que podem ser disparadas de acordo com certas condições. Estas regras incluem validações e descrições explícitas sobre algum domínio.

Os *scripts* são estruturas que representam uma sequência de eventos realizados para a resolução de um problema do passado. Um agente segue um *script* ou um *script* pode realizar um histórico das atividades dos agentes do sistema. Esta é a forma proposta pela arquitetura para implementar a abordagem baseada em casos.

Uma interessante característica da arquitetura proposta é a flexibilidade oferecida. O assistente poderia ser aplicado a diversos tipos de domínio, promovendo sua reusabilidade.

3.7.3 Arquitetura proposta por Genesereth

O trabalho de Genesereth e Ketchpel (1994) está focado na interoperabilidade de software. A proposta foi facilitar a criação de sistemas com habilidade de interação através de uma engenharia de software baseada em agentes. Nesta abordagem, os programas de aplicação são escritos como sendo agentes de software. Estes agentes podem ser interpretados como componentes que comunicam-se com seus pares por meio da troca de mensagens através de uma linguagem de comunicação de agentes. Apesar da semelhança com objetos, os agentes utilizam uma linguagem comum com uma semântica independente de agente enquanto que mensagens entre objetos podem variar de objeto para objeto. Esta arquitetura adota a abordagem ACL (*Agents Communication Language*), apresentada na seção 2.6, para a comunicação entre os agentes.

Uma questão também tratada por Genesereth e Ketchpel foi a preocupação com os programas já existentes. Sendo assim, eles propõem três abordagens para o que denominaram **agentificação**:

- Implementar um programa tradutor que atua como um mediador entre o programa e outros agentes.
- Implementar uma camada adicional para o programa existente, provendo este com a capacidade de se comunicar via ACL.
- Reescrever o programa original. Esta proposta deve ser utilizada apenas em último recurso.

Uma vez que a arquitetura tem definida sua plataforma de comunicação, torna-se importante entender como os agentes estão organizados. Existem duas abordagens diferentes: comunicação direta e coordenação assistida. Na comunicação direta, a coordenação é gerenciada pelo próprio agente. Na coordenação assistida, o agente delega a atividade de coordenação à programas especiais.

A principal vantagem da comunicação direta está na não dependência de outros programas. Duas arquiteturas conhecidas que implementam a comunicação direta são as redes de contrato (*contract net*) e compartilhamento de especificação (*specification sharing*). Nas redes de contrato, um agente solicita que outros agentes submetam suas propostas, os quais declaram-se para a realização de certa atividade. O agente recebe e analisa as declarações recebidas e estabelece um contrato com agente mais promissor. O compartilhamento de especificação usa a abordagem inversa, onde os agentes divulgam seus interesses e capacidades para os demais agentes. Neste caso, o número de mensagens trocadas é menor que nas redes de contrato, pois não há necessidade de haver sempre uma negociação.

O maior problema destas abordagens é a quantidade de tráfego gerado, uma vez que são soluções baseadas na difusão de mensagens. A complexidade do processo de negociação também é repassada aos agentes. Sendo assim, é proposta uma solução baseada em comunicação indireta denominada sistema federado.

No sistema federado, a interação dos agentes é assistida por um programa especial denominado facilitador, o qual oferece um conjunto de serviços de coordenação. A figura 6 apresenta uma arquitetura baseada em um sistema federado.

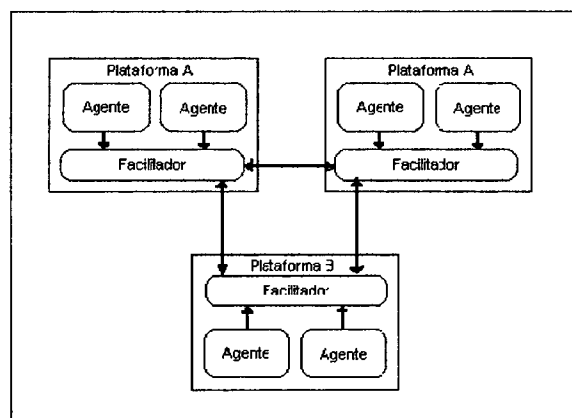


Fig. 6 – Federação de agentes.

Nesta proposta, os agentes utilizam uma estrutura semelhante ao compartilhamento de especificação, mas apenas com seu facilitador. O facilitador atua então como um mediador, roteando mensagens (solicitações e respostas) de acordo com seu conhecimento interno.

A principal característica dos facilitadores que o diferenciam de simples roteadores de mensagens (*brokers*) está associada à sofisticação de processamento. Um facilitador pode utilizar a documentação das necessidades e capacidades de um agente expressada através da ACL para melhorar o processo do roteamento de mensagens.

4 Raciocínio Baseado em Casos

4.1 Introdução

Os Sistemas Especialistas (SE) representam um dos grandes sucessos das pesquisas da Inteligência Artificial (IA). Este sucesso deve-se não apenas às características de inteligência que estes sistemas apresentam, mas por sua grande abrangência de aplicação. Entretanto, existem alguns problemas conhecidos:

- **Criatividade:** SE's não são preparados para responder a situações não esperadas.
- **Aprendizado:** SE's não possuem capacidade para aprender de forma autônoma, sendo necessário atualizações explícitas.
- **Analogia:** SE's não são capazes de raciocinar utilizando experiências passadas, sendo altamente dependentes da informação simbólica de entrada.
- **Limitação:** SE's não são muito bons para reconhecer as suas limitações, ou seja, quando um problema está fora de seu domínio de conhecimento.

Watson (1997) reafirma o sucesso dos SE's, mas também aponta algumas dificuldades encontradas pelos pesquisadores desta área. Segundo este autor, o processo de recuperação do conhecimento é complicado, tornando sua implementação difícil. Além disso, SE's tendem a se tornar lentos e incapazes de manipular grandes volumes de informação. Por fim, a inabilidade de aprender automaticamente torna o processo de manutenção difícil.

Para resolver a lacuna deixada pelos sistemas especialistas, os pesquisadores de IA têm explorado soluções na cognição humana. Segundo Sun (1996), o raciocínio é um aspecto fundamental da cognição humana. Ele está envolvido em todos os tipos de processos cognitivos, variando do entendimento da linguagem ao processo de tomada de decisão. O processo de reutilizar episódios passados corresponde a uma forma frequente e poderosa do raciocínio humano. Para Aamodt e Plaza (1994), esta alegação está apoiada na pesquisa da psicologia cognitiva, onde vários estudos demonstram o uso de experiências de situações prévias no processo humano de aprendizado e resolução de

problemas. Neste sentido, o Raciocínio Baseado em Casos (RBC) é uma técnica de IA que baseia-se na premissa de que seres humanos utilizam um raciocínio analógico ou experimental para aprender e resolver problemas complexos. Sistemas RBC utilizam a filosofia de aprender através de situações que aconteceram no passado (aprendizado por experiência). Leake (1996) apresenta dois pressupostos para o emprego do raciocínio baseado em analogia: problemas similares têm soluções similares e os tipos de problemas se repetem.

As principais características que diferem os sistemas RBC dos sistemas especialistas são apresentadas em (Watson, 1997):

- Sistemas RBC não requerem um modelo explícito do domínio. A recuperação do conhecimento é feita através da recuperação de casos armazenados.
- As dificuldades de implementação são reduzidas. A tarefa é encontrar os aspectos relevantes que descrevem um caso, ao invés da criação de um modelo explícito.
- Sistemas RBC podem ser combinados com técnicas de banco de dados para gerenciar grandes volumes de informação.
- A manutenção do conhecimento é simplificada pela capacidade que os sistemas RBC possuem de aprender novas informações na forma de casos.

Kolodner e Leake (1996) acrescentam que sistemas RBC oferecem também tempo de resposta rápido e a possibilidade de trabalhar em domínios que não são completamente conhecidos. Estas características possibilitam a aplicação do raciocínio baseado em casos em diversos tipos de tarefas. Algumas das possíveis aplicações são:

- **Sistemas de diagnóstico:** estes sistemas procuram recuperar casos acontecidos que tenham similaridades na sua lista de sintomas com um novo caso e sugerem diagnósticos baseados nos casos que melhor se encaixam. A maior parte dos sistemas instalados são deste tipo, sendo muitos na área médica.
- **Sistemas de *help desk*:** os sistemas de diagnóstico podem ser aplicados na área de atendimento a clientes, lidando com problemas em produtos ou serviços.
- **Sistemas de avaliação:** estes sistemas são utilizados para determinar valores para variáveis, comparando-as com o valor conhecido de um caso similar. Este tipo de aplicação é muito comum nas áreas de finanças e *marketing*.

- **Sistemas de apoio à decisão:** estes sistemas lidam com problemas complexos oferecendo suporte para a recuperação de informações relevantes em problemas similares. Usualmente, esta recuperação é feita em documentos.
- **Sistemas de projeto:** arquitetos e engenheiros podem utilizar estes sistemas para recuperar casos anteriores que tenham características similares ao projeto em desenvolvimento. Outras técnicas de raciocínio podem ser combinadas, oferecendo suporte durante todas as etapas do projeto.

4.2 Histórico

No final da década de 70, Schank e Abelson (1977) propuseram que o conhecimento humano podia ser armazenado na forma de um conjunto de roteiros (*scripts*). Estes roteiros seriam descrições sobre situações vivenciadas que poderiam ser recuperados e utilizados para inferir novos roteiros. Este trabalho é considerado por muitos pesquisadores como sendo uma das principais origens do raciocínio baseado em casos (Aamodt e Plaza, 1994), (Watson, 1997).

Entretanto, Aamodt e Plaza (1994) consideram que o trabalho de Wittgenstein, em 1953, pode ter sido a base filosófica para o RBC. Wittgenstein observou que conceitos naturais, como mesas e cadeiras, são na verdade polimórficos e não podem ser classificados como um simples conjunto de facilidades suficientes e necessárias, mas através de um conjunto de instâncias (casos) que possuam similaridades.

Schank (1982) apresentou seus estudos sobre memória dinâmica e como a manipulação de casos passados e padrões de situação poderiam ser aplicados à resolução de problemas e ao aprendizado. Schank definiu que um padrão de situação poderia ser descrito através de estruturas, as quais ele denominou de pacotes de organização de memória (MOP – *Memory Organization Packets*). Watson (1997) lembra que o modelo teórico para analogia proposto por Gentner (1983) também foi de grande relevância para área de RBC.

Janet Kolodner também é umas das precursoras em RBC. Um dos primeiros sistemas que utilizou esta abordagem foi o CYRUS (Kolodner, 1983). Este sistema possuía um modelo de memória baseado no modelo de memória dinâmico (MOP) proposto por

Schank. Foi utilizada uma estrutura hierárquica chamada pacotes de organização de memória episódica, onde casos específicos com propriedades similares eram organizados em estruturas mais generalizadas. O CYRUS era basicamente um sistema de perguntas e respostas sobre viagens e reuniões do Secretário de Estado Americano Cyrus Vance. Segundo Aamodt e Plaza (1994), o modelo de memória de casos utilizado para o desenvolvimento do CYRUS foi a base de muitos outros sistemas RBC.

Outro trabalho considerado muito relevante para o RBC foi desenvolvido por Bruce Porter e seus colegas em 1986 (Aamodt e Plaza, 1994), (Watson, 1997). Porter aplicou a abordagem do aprendizado por conceitos para classificação de tarefas. Este trabalho serviu como base para o desenvolvimento do sistema PROTOS. Este sistema enfatizava a integração do conhecimento geral sobre o domínio com o conhecimento específico de casos em uma única estrutura de representação de conceitos baseada em exemplos.

Desde a década de 90, RBC tem sido um campo de grande interesse. Esta motivação não é meramente acadêmica, mas também possui um grande apelo comercial. Diversas ferramentas e aplicações já estão disponíveis e em uso. Raciocínio baseado em casos é freqüentemente utilizado como um termo genérico para descrever técnicas que utilizam raciocínio por analogia. Segundo Watson (1997), Estados Unidos e Alemanha tem se destacado nesta área.

4.3 Definição

Como apresentado, o raciocínio baseado em casos é uma área relativamente recente. Mesmo assim, existe um grande interesse pelas possibilidades que esta abordagem compreende. Diferentemente da teoria sobre agentes, apresentada no capítulo anterior, os pesquisadores em RBC não divergem muito sobre sua definição. A maioria dos autores concordam que o RBC é um método de raciocínio baseado na proposta de utilizar experiências passadas encapsuladas em estruturas de dados como a base para lidar com novas situações similares. A abordagem parece ser intuitiva: quando uma nova situação acontece, deve-se tentar alguma coisa que já foi utilizada com sucesso. Usualmente, as soluções utilizadas em situações similares devem ajudar na solução do novo problema. É importante notar que o raciocínio pode funcionar também por contra-

exemplos. Neste caso, dada uma nova situação, deve-se procurar descartar aquelas soluções utilizadas em situações similares que resultaram em insucessos.

Segundo Gentner (1983), o raciocínio baseado em casos é um novo ramo de resoluções de problemas por analogia que utiliza a experiência para resolver novos problemas. O processo de raciocínio por analogia identifica determinados aspectos em novos e antigos problemas e procura utilizar as soluções encontradas para inferir uma nova solução.

Riesbeck e Schank (1989) definem que um sistema RBC resolve novos problemas adaptando soluções que solucionaram problemas do passado.

Aamodt e Plaza (1994) descrevem RBC como sistemas capazes de utilizar um conhecimento específico de experiências passadas (casos) para resolver novos problemas. A solução é encontrada através da recuperação de um caso similar no passado.

Leake (1996) argumenta que o raciocínio utilizado em RBC está baseado em lembrança. A fonte primária do conhecimento em um sistema RBC está na sua memória de casos armazenados que caracterizam episódios específicos do passado. Sendo assim, novas soluções são geradas pela recuperação dos casos mais relevantes na memória e pela adaptação destes casos de forma a atenderem as novas necessidades.

Watson (1997) define RBC como um paradigma de resolução de problemas que envolve a aproximação entre o problema atual e um problema resolvido com sucesso no passado. Este processo pode ser otimizado adaptando soluções que mais acuradamente se aproximam do problema atual.

Apesar de sua definição genérica, Aamodt e Plaza (1994) consideram também que o paradigma RBC abrange uma faixa de diferentes métodos para manipular a informação contida nos casos. Sendo assim, eles apresentam alguns outros termos e definições encontrados na área:

- **Raciocínio baseado em exemplos:** nesta abordagem, um conceito é definido, por extensão, como um conjunto de seus exemplares. A tarefa corresponde a um processo de classificação dos exemplos. A modificação de uma solução encontrada (processo de aprendizado) está fora do escopo desta abordagem.

- **Raciocínio baseado em instâncias:** esta abordagem é uma especialização do raciocínio baseado em exemplos, onde, para a definição de um conceito, deve-se ter um número relativamente grande de instâncias. A representação das instâncias é usualmente simples, uma vez que o principal foco é o estudo do aprendizado automatizado sem intervenção humana.
- **Raciocínio baseado em memória:** esta abordagem considera uma coleção de casos como sendo uma grande memória, e o raciocínio como um processo de acessar e procurar nesta memória. A diferença para outros métodos RBC está no uso de técnicas de processamento paralelo.
- **Raciocínio baseado em casos:** apesar desta abordagem ser usada como um termo genérico, o método típico de raciocínio baseado em casos possui algumas características que o distinguem dos demais métodos apresentados. Neste método, um caso tem um certo grau de riqueza nas suas informações e um certo tipo de complexidade com relação a sua organização interna. Além disto, outra característica importante refere-se a capacidade de modificar ou adaptar uma solução recuperada quando esta é aplicada em um contexto de resolução de problemas diferente. Este método também utiliza um *background* de conhecimentos gerais, com representações explícitas de um contexto e das regras, nos quais os processos RBC variam.
- **Raciocínio baseado em analogia:** apesar deste termo ser freqüentemente usado como um sinônimo para RBC, esta abordagem está focada em métodos que resolvem novos problemas baseados em casos passados de um diferente domínio.

4.4 A Abordagem RBC

Os sistemas RBC utilizam um processo interativo constituído genericamente por: identificação da situação atual, busca da experiência mais semelhante na memória e aplicação do conhecimento desta experiência na situação atual. Entretanto, a literatura usualmente não considera a identificação da situação atual como parte do processo RBC, adotando um modelo genérico baseado em quatro etapas: recuperar, reutilizar, revisar e reter. Autores como Aamodt e Plaza (1994) e Watson (1997) referem-se a estas etapas como o ciclo do RBC (figura 7):

1. **Recuperar**: busca, na base de casos, um ou mais casos a partir da comparação da situação atual (caso de entrada) com cada um dos casos da base (casos candidatos). O processo de comparação é realizado através da avaliação das similaridades entre o caso de entrada e os casos candidatos.
 2. **Reutilizar**: procura utilizar a informação e o conhecimento contidos nos casos recuperados (pode ser apenas um) para resolver o caso de entrada.
 3. **Revisar**: procura avaliar a solução proposta.
 4. **Reter**: adiciona o caso de entrada com sua solução revisada na base de casos.
- Esta etapa representa a característica de aprendizagem de um sistema RBC.

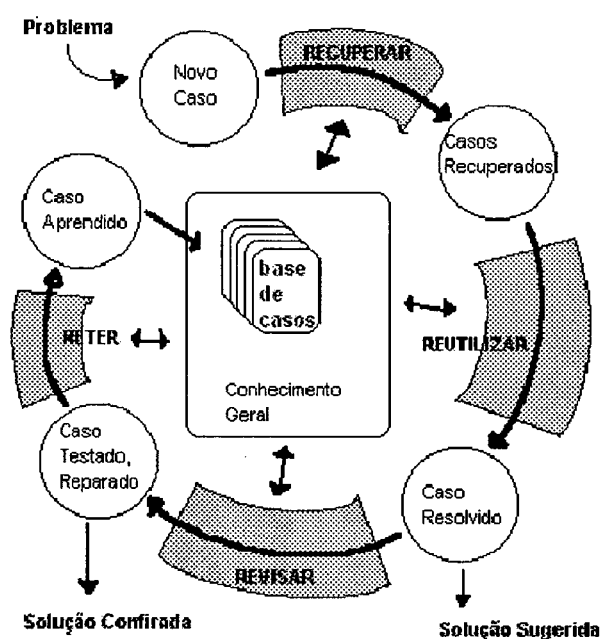


Fig. 7 – O Ciclo do RBC (Aamodt e Plaza, 1994).

Apesar do ciclo apresentado representar um grande número de sistemas RBC, autores como Kolodner e Leake (1996) consideram que um sistema RBC pode ter um processo mais detalhado: recuperar, propor, justificar ou adaptar, criticar, avaliar e armazenar. Esta proposta para o ciclo CBR é apoiada no fato que os sistemas RBC podem ser freqüentemente classificados de acordo com suas tarefas e objetivos (Leake, 1996). Sendo assim, os sistemas RBC podem ser divididos em sistemas interpretativos e de resolução de problemas.

4.4.1 Sistemas Interpretativos

Os sistemas interpretativos utilizam casos passados como pontos de referência para a classificação e caracterização de novos casos. A classificação da nova situação é realizada através de uma comparação com os casos já classificados. O foco destes sistemas está na interpretação do novo caso, e não na resolução de um problema.

O ciclo destes sistemas caracteriza-se pela etapa de justificação. Após a recuperação dos casos similares e de uma interpretação inicial ter sido feita e proposta, inicia-se a etapa de criação de argumentos para justificar a solução proposta. A justificativa é feita através de uma análise balanceada das similaridades e diferenças com os casos recuperados. A etapa de crítica pode ser utilizada para testar a solução com sua justificativa, antes que a avaliação final seja feita.

Estes sistemas são particularmente importantes para aplicação no ensino. Interpretar um problema e justificar esta interpretação parece ser uma forma interessante de construir conhecimento sobre um domínio. Na verdade, o uso de casos não precisa ocorrer apenas de forma automática através de um sistema computacional. Eles podem ser utilizados por alunos em uma sala de aula. Neste caso, o professor pode orientar o processo RBC de forma com que os alunos sejam os executores do ciclo. Eventualmente, ferramentas computacionais podem auxiliar os alunos neste processo.

4.4.2 Sistemas de Resolução de Problema

Os sistemas de resolução de problema utilizam as soluções adotadas em casos passados para sugerir soluções que podem ser aplicadas a novas situações. O enfoque destes sistemas está na adaptação das soluções recuperadas para gerar a solução de um novo problema. A adaptação substitui a etapa de justificativa dos sistemas interpretativos, permitindo uma revisão do caso recuperado no sentido de ajustar ao caso de entrada. A etapa de crítica pode fazer com que o processo de adaptação tenha que ser revisto antes da solução ser aplicada, indicando uma recursividade inerente.

4.5 Representação do Conhecimento

A representação de conhecimento pode ser interpretada como sendo um conjunto de métodos que modelam um conhecimento especialista para disponibilizá-lo em um sistema inteligente. Apesar de um sistema RBC ser extremamente dependente da estrutura e do conteúdo de sua base de casos, seu conhecimento não está presente apenas nos seus casos (memória), mas também nas suas etapas de desenvolvimento. Webber (1998) lembra que esta é uma importante distinção do raciocínio baseado em casos quando comparado a outras técnicas da IA, e apresenta algumas evidências desta conclusão:

- Estabelecer o que deve ser considerado um caso depende do conhecimento de um especialista sobre o domínio em questão.
- A forma com que os casos são organizados na base de casos também utiliza o conhecimento sobre o domínio.
- A identificação sobre o que faz um caso ser similar a outro é totalmente dependente de domínio.
- O ajuste da situação ou adaptação também deve ser desenvolvido com conhecimento especialista.

4.6 Processo de Desenvolvimento RBC

O processo de desenvolvimento RBC é uma tarefa que pode variar dependendo do tipo de sistema em questão. Entretanto, pode-se definir uma metodologia genérica que compreende uma sequência de etapas. Estas etapas são problemas a serem modelados que, quando resolvidos, compreendem um sistema RBC. Estas etapas são:

1. Representação dos casos.
2. Indexação.
3. Recuperação dos casos.
4. Adaptação.
5. Aprendizagem.
6. Análise da situação.

É importante notar que este processo de desenvolvimento não aborda a formulação inicial para um sistema RBC, que é a identificação da situação atual ou do caso de entrada. Entretanto, pode-se subentender que esta identificação está inserida na própria definição de um caso no sistema. Outro ponto importante a ser observado é que algumas das etapas descritas podem não ser necessárias, dependendo do domínio para o qual o sistema será desenvolvido. Como já discutido anteriormente, não é obrigatório que um sistema RBC possua capacidade de adaptação, ou mesmo a análise da situação.

4.6.1 Representação dos Casos

A principal parte do conhecimento nos sistemas RBC está representada através de seus casos. Um caso pode ser entendido como a abstração de uma experiência descrita em termos de seu conteúdo e contexto, podendo assumir diferentes formas de representação. Aamodt e Plaza (1994) definem um caso como sendo uma situação de problema. Watson (1997) interpreta um caso como uma parte de conhecimento contextualizada representando uma experiência. O conjunto destes casos compreendem uma base de casos. Kolodner (1993) acrescenta que os procedimentos de acesso aos casos também fazem parte da base de casos.

A representação dos casos é uma tarefa complexa e importante para o sucesso do sistema RBC. Inicialmente, o problema é decidir o que será armazenado em um caso e encontrar a estrutura mais apropriada para descrever seu conteúdo. Definido o caso, resta a questão sobre como os casos serão organizados e indexados para efetivar sua recuperação e reutilização (Aamodt e Plaza, 1994).

Um caso deve possuir dois componentes básicos: a descrição do problema e a descrição da solução. Watson (1997) afirma que um caso contém uma lição passada que é o conteúdo do caso e o contexto no qual a lição pode ser utilizada, sendo representado por um registro que apresente tipicamente: **problema** que descreve o estado do contexto onde o caso ocorreu, e **solução** que expõe a solução derivada daquele problema. Kolodner (1993) inclui ainda um terceiro componente: o resultado da aplicação da solução ao problema. Os aspectos desta definição abordam mais diretamente os sistemas de resolução de problema. Em sistemas interpretativos, o enfoque é a representação de experiências de forma a facilitar a interpretação das mesmas.

A IA oferece uma grande variedade de formalismos de representação que podem ser utilizados para modelar o conteúdo de um caso computacionalmente:

- **MOP's:** os pacotes de organização de memória (MOP – *Memory Organization Packages*) permitem a representação de eventos estereotipados (Schank, 1982). MOP's são organizados em estruturas que agrupam eventos similares através de uma hierarquia todo-parte. MOP's são estruturas que mapeam experiências através de cenários que incluem situações representadas através de informação normativa e descritiva.
- **Redes Semânticas:** consistem de objetos (nós) interligados através de conexões (arcos) que descrevem as relações entre estes objetos. Os nós representam objetos, conceitos e eventos. Os arcos são direcionados e representam relações e atributos.
- **Frames:** são estruturas de dados propostas por Minsky (1975) que representam uma entidade através de suas características e capacidades. Apesar de sua semelhança inicial com as redes semânticas, os frames diferem na forma de representação da entidade. As características são representadas por pares atributo-valor e as capacidades são representadas por métodos. Da mesma forma que na abordagem orientada a objetos, os frames podem ser organizados em estruturas hierárquicas de especialização e todo-parte.
- **Regras:** são estruturas compostas de duas partes: premissas e conclusões. As premissas são um conjunto de expressões que avaliam a presença ou não de determinados fatos. As conclusões são um conjunto de expressões que modificam fatos existentes ou inserem novos fatos. O processo de disparo de uma regra é simples em sua definição: quando todas as premissas são satisfeitas, a conclusão pode ser executada. Quando há mais de uma regra apta ao disparo, devem-se utilizar mecanismos de avaliação eficientes e usualmente dependentes de domínio para descobrir qual a regra mais relevante a ser utilizada.
- **Formulários:** um formulário é um conjunto de campos valorados semelhantes a registros em um banco de dados. Kolodner (1993) apresenta este formalismo como uma alternativa para representar casos em estruturas organizacionais planas. Os campos de um formulário são denominados descritores. Descritores são pares atributo-valor que caracterizam a informação contida em um caso.

4.6.2 Indexação

A indexação é um problema central do raciocínio baseado em casos, envolvendo a determinação dos tipos de índices que serão utilizados pela etapa de recuperação. Da mesma forma que índices permitem acelerar a busca em bancos de dados, eles são utilizados no sistemas RBC para acelerar a recuperação de casos. Existem dois tipos de informação dentro de um caso (Watson, 1997):

- Informação indexada utilizada para recuperação.
- Informação não indexada que oferece informação contextual de valor para o usuário, mas não usada diretamente na recuperação.

De acordo com Kolodner e Leake (1996), os índices de um caso são combinações de importantes descritores, capazes de estabelecer distinções entre os casos existentes. O algoritmo de recuperação utiliza estes índices para selecionar os casos conforme suas necessidades. Webber (1998) reafirma a importância da indexação na avaliação de similaridade. O conjunto de descritores utilizados como índices permitirão definir o que faz um caso ser similar a outro, representando a relevância dos casos.

Kolodner (1993) e Watson (1997) apresentam características desejáveis para os índices:

- Índices devem ser vaticinantes, permitindo predizer informações sobre os casos.
- Índices devem ser abstratos o suficiente para permitir uma maior abrangência do uso da base de casos.
- Índices devem ser concretos o suficiente para serem reconhecidos.
- Índices devem endereçar os propósitos de utilização dos casos.

Webber (1998) afirma que o processo de indexação pode oferecer a oportunidade de superar deficiências na descrição do caso e transformá-lo em informação útil para o sistema.

4.6.3 Recuperação dos Casos

Dado um problema a ser resolvido (caso de entrada), a etapa de recuperação realiza a busca na base de casos e seleciona quais casos podem ser aproveitados. O processo de busca é feito por algoritmos que selecionam casos com similaridade quando comparados ao caso de entrada. As tarefas que compreendem a recuperação de casos

são: métrica da similaridade, recuperação de casos e seleção do caso escolhido. Aamodt e Plaza (1994) descrevem que a tarefa de recuperação de casos inicia com a descrição de um problema e termina quando um caso mais similar é encontrado.

A métrica de similaridade é uma função que permite avaliar analiticamente os graus de similaridade entre dois casos. Usualmente, são atribuídos pesos diferentes a cada uma das características de um caso e métodos de agregação como a média ponderada são utilizados. O valor de cada peso é diretamente proporcional à importância de cada característica.

A recuperação é uma função que retorna os casos mais similares. Usualmente, é utilizado um limiar de similaridade, que orienta o processo no sentido de recuperar apenas aqueles casos que tenha similaridade superior àquele limiar. Pode ser utilizado também um limite no número de casos a ser retornado.

O processo de seleção do caso mais relevante é a última tarefa da etapa de recuperação de casos. O sistema pode implementar o método de seleção usando de várias formas: perguntando ao usuário, combinação de heurísticas ou aplicação de regras.

4.6.4 Adaptação

Após a recuperação do caso mais similar, torna-se necessário revisar este caso e sua solução para verificar a necessidade de adaptação ao problema de entrada antes de utilizá-lo. A revisão de casos consiste em avaliar a solução do caso recuperado e verificar as seguintes possibilidades (Aamodt e Plaza, 1994): se a solução obteve êxito, aprender com o sucesso, caso contrário, reparar a solução utilizando conhecimentos específicos do domínio do problema.

A tarefa de avaliação está relacionada com a análise dos resultados obtidos pela aplicação da solução no ambiente real (perguntando a um instrutor ou realizando a tarefa no mundo real). Este processo geralmente ocorre fora do sistema RBC e os resultados da aplicação de uma solução podem demorar a aparecer, dependendo do tipo da aplicação. A reparação de um caso envolve a detecção dos erros ocorridos na solução de um caso, e a recuperação ou geração de explicações para estes erros.

4.6.5 Aprendizagem

Uma característica muito interessante que os sistemas RBC podem apresentar é a capacidade de aprendizagem. É importante lembrar que o raciocínio baseado em casos originou-se em estudos sobre máquinas de aprendizado. A aprendizagem em sistemas RBC pode ser empregada ao nível dos casos e da base de casos, denotando um paradigma capaz de aprender a partir da atualização da sua base de casos após um problema ter sido resolvido.

A retenção de casos corresponde ao processo de incorporar à base de casos, informações úteis relativas à resolução de um novo problema. Este processo corresponde à aprendizagem de um sistema RBC, sendo disparado pelas tarefas de avaliação e adaptação de soluções (Aamodt e Plaza, 1994).

Segundo Kolodner e Leake (1996), a efetivação do aprendizado também pode ocorrer como o resultado da atualização do seu raciocínio. O mecanismo de raciocínio pode ser mais eficiente quando está dotado de maior capacidade de relembrar soluções passadas, as quais serão adaptadas para o novo caso, derivando-se novas soluções a cada momento. Ele passa a ser mais competente quanto maior for sua capacidade de derivar soluções melhores, quando desprovido de experiência sobre o problema.

O processo de aprendizado em um sistema RBC não deve ser dirigido apenas pelo sucesso da aplicação de um caso recuperado em um caso de entrada. É importante também aprender com os fracassos. Há dois motivos para aprender com fracassos: soluções falhas revelam a necessidade do aprendizado e revelam ao sistema o que deve ser aprendido (Leake, 1996).

4.6.6 Análise da Situação

Os sistemas RBC podem funcionar perfeitamente apenas com as etapas discutidas até o momento. Entretanto, existe uma questão problemática a ser tratada que é quando o usuário tem dificuldade para apresentar o caso de entrada no formato escolhido para representar a base de casos. Deve-se entender que este formato não precisa ser a estrutura interna de armazenamento de casos, mas a filosofia embutida nesta representação. Sendo assim, torna-se necessário haver uma etapa de ajuste da situação

inicial, com o objetivo de modelá-la em um caso de entrada que possa ser identificado pelo sistema para o processo de recuperação.

4.7 Ensino Baseado em Casos

O capítulo 2 abordou algumas características do uso de IA em educação. Foi visto que em sistemas computacionais de apoio ao ensino tradicionais, o projetista usualmente possui conhecimento sobre o domínio e decide sobre como este conhecimento será disponibilizado. O desenvolvimento de tais sistemas é realizado através de análises das tarefas que compreendem o domínio e das experiências dos professores. O projetista, de posse destas informações, define como o conhecimento será repassado aos estudantes: sequência, nível de dificuldade e forma de apresentação. A existência de um modelo previamente definido do domínio faz com que esta abordagem não esteja preparada para lidar com situações inesperadas (Khan e Yip, 1996).

Como discutido anteriormente neste capítulo, uma das vantagens de aplicação do raciocínio baseado em casos está em trabalhar com conhecimento do domínio parcialmente incompleto. Não há a necessidade que um modelo explícito do domínio esteja presente. Esta característica permite que o crescimento do conhecimento seja incremental. Além disso, no enfoque educacional, os casos podem ser explorados como situações a serem apresentadas aos estudantes para que estes tentem encontrar soluções adotadas anteriormente em problemas semelhantes, sintetizá-las, aplicá-las na nova solução e ainda fornecer as explicações que motivaram a escolha. Estes aspectos fazem com que o uso de casos seja interessante como uma ferramenta no processo de ensino/aprendizagem.

Khan e Yip (1996) apresentam 14 princípios pedagógicos que têm contribuído para o desenvolvimento de sistemas de ensino baseado em casos:

1. **Ensino baseado em estórias:** explora o interesse inerente dos estudantes de aprender através de estórias e o desejo básico de professores e especialistas por contar estórias que encapsulam suas experiências.

2. **Ensino auto direcionado:** os estudantes são motivados a refinar seus modelos cognitivos de um domínio por auto exploração de um ambiente. Este ambiente deve permitir que os modelos possam ser testados.
3. **Instrução significativa:** estórias são melhor apresentadas em um contexto que habilite o estudante determinar onde ele está no conteúdo e como ele poderia se conectar a outras estórias.
4. **Ensino dirigido ao impasse:** estórias poderiam ser utilizadas para ilustrar pontos pedagógicos somente quando o estudante necessita saber a informação.
5. **Instrução centrada na tarefa:** habilidades devem ser ensinadas em tarefas onde o conhecimento é normalmente aplicado.
6. **Ensino dirigido a falha:** estudantes deveriam ser motivados a aprender a partir de situações de falha durante a execução de uma tarefa.
7. **Ensino dedutivo:** pessoas aprendem sobre um domínio deduzindo regras generalizadas a partir de casos dados. Portanto, o ensino pode ocorrer através da apresentação de exemplos bem escolhidos.
8. **Congruência instrucional:** uma seleção conduzida de exemplos assegura a realização de metas instrucionais pretendidas e evita erros de entendimento.
9. **Raciocínio analógico:** estudantes utilizam a lembrança de soluções passadas para resolver novos problemas. Estas soluções podem ainda ser generalizadas para serem aplicadas em outros domínios. Esta análise pode ser considerada o processo de entendimento por parte do estudante.
10. **Estratégias de elaboração:** estudantes podem aprender a criar suas próprias explicações se eles aprendem boas estratégias para elaborar o conteúdo dos exemplos trabalhados.
11. **Auto explicação:** estudantes aprendem através da construção de explicações que os ajudam a entender o conteúdo. Desta forma, o entendimento do estudante pode ser testado pela análise de suas explicações.

12. **Perguntas explicativas:** estudantes aprendem através de respostas dadas a um conjunto de perguntas investigativas.
13. **Explicações derivativas:** professores incorporam suas explicações passadas na derivação de novas explicações.
14. **Auxílio à memória:** humanos funcionam melhor quando são assistidos por uma memória externa que os auxilia com raciocínio analógico.

Alguns destes princípios estão claramente presentes nos trabalhos de Schank et al (1994) quando ele explora o raciocínio baseado em casos para a construção de explicações. Outros pesquisadores do grupo de Schank também exploram o uso de casos como uma abordagem para modelar ensino baseado em histórias (Ferguson et al, 1992), (Burke e Kass, 1996).

O próximo capítulo apresenta um modelo para ensino colaborativo utilizado para o ensino a distância, onde procura-se atender alguns dos princípios pedagógicos discutidos nesta seção e no capítulo 2.

5 Modelo para Ensino Colaborativo

5.1 Introdução

Colaboração é uma parte essencial para muitos tipos de trabalho, mas não é ainda muito utilizada nas escolas. De acordo com Crook (1994), existe uma demanda pela modelagem de ambientes de ensino colaborativo para uso através de redes de computadores. Ambientes de ensino colaborativo são ambientes eletrônicos que suportam e mediam trabalho e aprendizado cooperativo em uma rede de computadores (Ayala e Yano, 1995).

Atualmente existem vários trabalhos em ambientes colaborativos. Cargile (1997) apresenta um ambiente virtual de ensino colaborativo chamado *Collaboratory*. Este ambiente é implementado como uma sala de aula virtual onde um grupo de estudantes interage através de um conjunto de ferramentas que permitem o ensino através da prática. Entretanto, apesar do *Collaboratory* possuir uma interface realmente interessante, baseada em ambientes 3D e 2D (mas não imersiva), ele não oferece tutores inteligentes. A responsabilidade de avaliar e mediar o trabalho é de professores humanos e dos próprios estudantes. Em (Oliver et al, 1997), é proposto ambiente de ensino colaborativo através da Internet. Este modelo discute como a utilização de ferramentas, como salas de discussão e páginas www (*world wide web*), pode ser aplicada para a colaboração dos estudantes. Novamente, este ambiente procura oferecer recursos para a colaboração, mas não são apresentados mecanismos para automatizar aspectos de tutoria inteligente.

Os agentes inteligentes também têm sido utilizados para a construção de ambientes de ensino colaborativo (Boy, 1997). Ayala e Yano (1994, 1995) apresentam um sistema de ensino colaborativo para o aprendizado da língua japonesa. Uma das características deste ambiente é ter sido desenvolvido para ser utilizado em um laboratório de computadores com os alunos trabalhando ao mesmo tempo. Outra questão é que sua interface foi projetada para uma aplicação específica. Boy (1997) apresenta como os diversos trabalhos de Schank em arquiteturas de ensino (Schank, 1982, 1991, 1994)

oferecem uma base para a construção de agentes em ambientes colaborativos. Os trabalhos de Schank estão focados principalmente na idéia de que o ensino pode ser alcançado pela exploração de casos e resolução de problemas. Usando esta mesma abordagem, Ploetzner e Fehse (1998) apresentam um modelo cognitivo para o ensino colaborativo através da troca de explicações e experiências entre os estudantes.

5.2 Definição do Modelo

Como visto, agentes inteligentes e raciocínio baseado em casos são abordagens que oferecem uma grande contribuição para a implementação de ambientes colaborativos. Da mesma forma, o uso da Internet oferece um suporte de comunicação ideal, pelas suas características de abrangência e baixo custo.

A análise dos trabalhos discutidos anteriormente mostra que eles apresentam diversas características interessantes para um sistema colaborativo. Entretanto, os sistemas não apresentam soluções integradas que permitam uma evolução do processo de ensino/aprendizagem tradicional para os novos paradigmas da educação. Algumas ferramentas conseguem trabalhar vários princípios pedagógicos da colaboração, mas não implementam modelos de orientação inteligente. Outras ferramentas são totalmente orientadas a um específico domínio de conhecimento.

Dentro deste contexto, este trabalho estabelece um modelo para o ensino colaborativo através da Internet. O objetivo principal é agregar diversos aspectos pedagógicos do processo colaborativo com o ensino através de experiências e por demanda. O modelo ainda permite que a cooperação possa ocorrer em modo assíncrono.

É definida uma plataforma orientada a agentes para suportar a cooperação de usuários através de uma rede de computadores. Torna-se importante ressaltar que os agentes podem se comunicar em qualquer rede baseada nos protocolos da Internet. Estes protocolos são adotados pela grande maioria das soluções em redes de computadores. Esta característica faz com que a cooperação possa ocorrer tanto em pequenos laboratórios de aula como através da Internet, sem limite de distância.

A plataforma define um sistema multi-agente que adota a estrutura de uma federação de agentes. A comunicação dos agentes utiliza uma estrutura formalizada baseada na

Linguagem de Comunicação de Agentes (ACL – *Agent Communication Language*). Como apresentado no capítulo 3, um sistema multi-agente é composto de agentes que compartilham um ambiente comum. Conseqüentemente, eles devem atuar coletivamente para identificar e resolver conflitos, sendo, ao mesmo tempo, beneficiados pela ação de outros agentes.

São definidas três classes de agentes para a implementação da plataforma: agentes de interface, agentes de informação e agentes consultivos. Esta abordagem foi inicialmente proposta em (Thiry, 1998a). Uma extensão desta abordagem para a aplicação no ensino à distância foi proposta em (Thiry, 1998b). Os agentes utilizam técnicas diferentes para resolver problemas específicos e interpretar situações. As principais abordagens de raciocínio utilizadas são raciocínio baseado em casos (o enfoque é ensino baseado em casos) e regras de produção.

O trabalho mostra a utilização desta plataforma como a base para a implementação de uma aplicação para suportar o ensino à distância. Esta ferramenta é desenvolvida como um navegador Internet inteligente.

5.3 Plataforma Multi-Agentes

O modelo genérico da plataforma possui três tipos de agentes assim definidos: agente de interface, agente de informação e agente consultivo (Fig. 8). O agente de interface é responsável pela interação com usuário. O agente de informação é relacionado com o acesso ao banco de dados e à base de conhecimento. O agente consultivo oferece assistência pessoal ao usuário. Os agentes cooperam através de uma rede de computadores, que pode ser a Internet, uma Intranet ou até mesmo uma rede local isolada.

A complexidade da plataforma proposta e dos diferentes tipos de serviços que os agentes podem oferecer indicam a necessidade da aplicação de mais de uma técnica de inteligência artificial. Embora tenha se pensado inicialmente na utilização de sistemas híbridos, os quais incorporam facilidades de duas ou mais técnicas na resolução de problemas (Thiry, 1998a), decidiu-se por utilizar as técnicas separadamente. Esta conclusão é motivada pelas diferentes aplicações das técnicas adotadas.

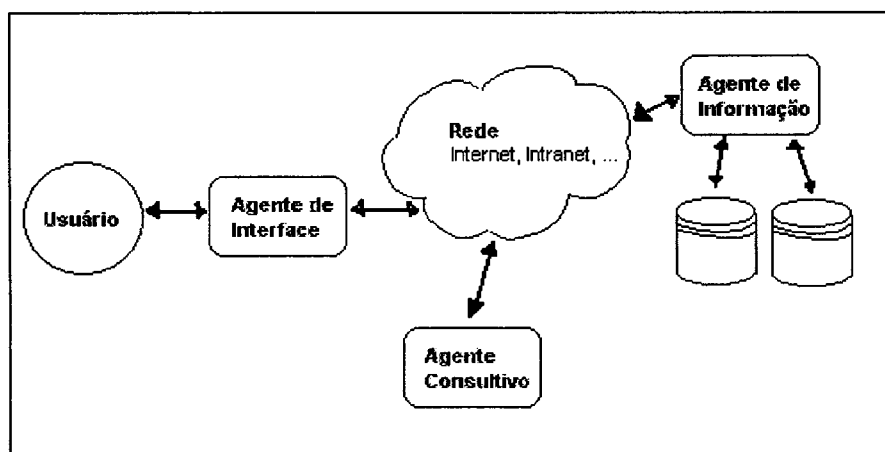


Fig. 8 – Arquitetura genérica para o ambiente de ensino colaborativo.

Inicialmente, dois paradigmas estão sendo combinados para orientar o processo de raciocínio dos agentes propostos: sistemas especialistas e raciocínio baseado em casos. Um sistema especialista é apropriado em domínios onde o conhecimento pode ser representado na forma de *frames* (objetos) e regras⁵. Como já discutido anteriormente, o enfoque adotado neste trabalho para a abordagem RBC é o ensino baseado em casos. Os casos passados auxiliam o usuário, informando sobre possíveis situações problemáticas e evitando repetição de erros.

5.3.1 Agente de Interface

O agente de interface interage com o usuário e coopera com os outros agentes trocando informações sobre capacidades, compromettimentos e metas de aprendizado dos usuários. Além disso, o agente de interface deve ser provido com facilidades para a representação de seu usuário. O agente de interface deve poder assumir o papel de seu usuário quando este não estiver presente. É importante notar que o agente de interface, em sua primeira versão, não oferece muitas funções para automatizar tarefas para seu usuário. Entretanto, eles foram estruturados para permitir a adição de funcionalidades de forma incremental.

⁵ A representação do conhecimento em sistemas especialistas pode utilizar outras abordagens, como redes semânticas. Entretanto, no contexto deste trabalho, sistemas especialistas utilizam basicamente regras de produção.

A figura 9 apresenta a estrutura básica de um agente de interface. É possível verificar quatro níveis bem estruturados: interface, interpretação, operação e comunicação.

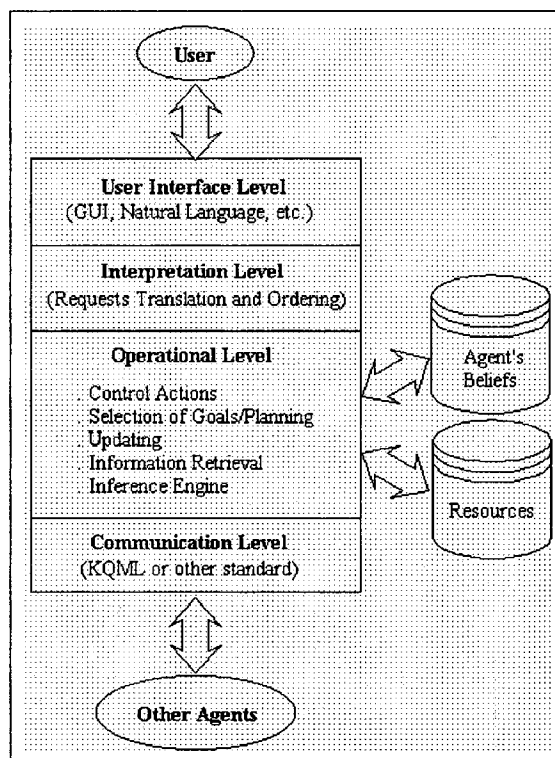


Fig. 9 – Estrutura básica do agente de interface.

O **Nível de Interface** oferece a funcionalidade de comunicação com o usuário. Ele é responsável por captar as consultas do usuário e por apresentar as mensagens vindas do sistema. O **Nível de Interpretação** tem a função básica de tradução. Ele procura preparar e ordenar as consultas feitas pelo usuário em uma estrutura de dados interna que é manipulada pelo agente. O **Nível Operacional** representa a parte principal do agente, pois é nele que reside a funcionalidade de raciocínio. Este nível é responsável pela monitoração das atividades do usuário e pela manutenção do conhecimento. O **Nível de Comunicação** é a parte do agente que lida com a troca de mensagens com outros agentes. O formalismo adotado neste nível para a representação das mensagens é a linguagem KQML.

Os agentes na arquitetura proposta estão sendo modelados através de uma estrutura de camadas. A proposta é oferecer funções bem definidas em cada camada. Esta analogia com arquiteturas de redes de computadores permite a independência de implementação para cada camada. A comunicação entre as camadas é realizada através de uma interface bem definida. Esta abordagem também oferece modularidade. Por exemplo, o **Nível de**

Interface está sendo inicialmente implementado através de menus *popup* e mensagens, mas uma interface em linguagem natural pode ser adicionada com impacto mínimo na estrutura do agente.

O processo de aprendizado do usuário consiste na criação de um conjunto de crenças sobre a informação disponível no sistema. Estas crenças estão sendo implementadas como duas bases de conhecimento diferentes. A primeira base utiliza uma abordagem para aprender e representar a informação sobre preferências do usuário. A segunda base de conhecimento utiliza raciocínio baseado em casos para a construção de uma representação parcial sobre o domínio, ou seja, o conhecimento a ser aprendido. O usuário pode solicitar ao agente de interface, a qualquer momento, a revisão dos casos que ele já tenha trabalhado. Torna-se importante ressaltar que, neste caso, não há necessidade de trocar mensagens com outros agentes. Todo o processo é feito localmente.

O banco de dados de recursos representa informação adicional. Ele armazena o endereço do agente, o nome como ele é conhecido e seu mapeamento na rede. Atualmente, está sendo implementado um histórico sobre tudo o que o agente de interface está fazendo, incluindo-se comunicação com usuário e com outros agentes. Esta facilidade permitirá, além de um acompanhamento dos resultados da implementação, uma interessante forma de avaliação da plataforma.

5.3.2 Agente de Informação

Este agente armazena a representação do conhecimento do domínio utilizada pelos agentes de interface como base para a construção de suas próprias interpretações do domínio. É definida uma interface de acesso aos dados. Os níveis de interpretação e interface não estão presentes, uma vez que o usuário não possui acesso direto a este agente. Os agentes realizam todo o processo de comunicação. Ambos, agentes de interface e agentes consultivos podem acessar o agente de informação. A figura 10 apresenta a estrutura básica do agente de informação.

A informação é dividida em duas categorias diferentes: material didático e base de conhecimento. O material didático compreende páginas em formato HTML (*Hypertext Markup Language*), imagens, textos e multimídia em geral. Esta categoria de

informação está sendo armazenada atualmente em um banco de dados relacional e através de um *site* Internet (páginas HTML). De fato, a melhor abordagem para armazenar o material didático seria um banco de dados orientado a objetos. Ganhos importantes com esta abordagem seriam um melhor armazenamento dos recursos de multimídia e a organização das páginas como objetos e não como arquivos em um conjunto de diretórios. Entretanto, a validação e a funcionalidade do modelo pode ser obtida com o uso de um banco relacional.

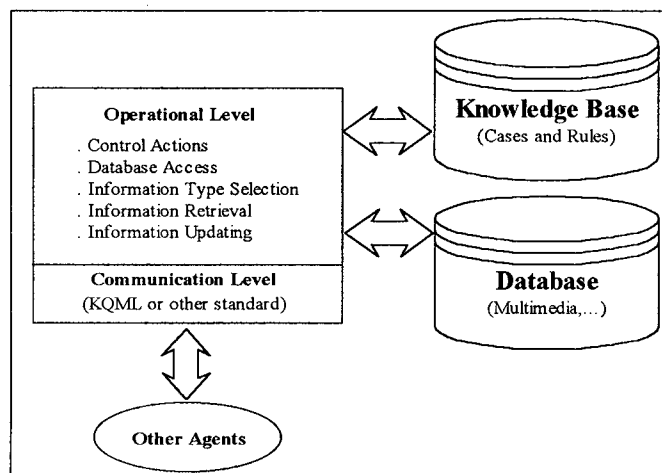


Fig. 10 – Agente de Informação.

A base de conhecimento é organizada através de um conjunto de casos e regras. Na verdade, a versão atual utiliza regras apenas para a estrutura de recuperação dos casos. Os casos representam as situações a serem exploradas pelos estudantes. Apesar deste trabalho diferir entre banco de dados e base de conhecimento, é importante ressaltar que os casos estão sendo armazenados também em um banco de dados relacional. A terminologia base de conhecimento continuará sendo adotada para identificar claramente o tipo de informação que está sendo discutida.

Os casos estão sendo modelados através de um formulário com os seguintes atributos:

- **Vocabulário:** define a matéria ou curso que está sendo ministrado. Um exemplo seria "Engenharia de Software".
- **Contexto:** estabelece o tipo de informação que o caso representa. Valores possíveis seriam "definição", "situação", "problema" e outros.

- Tema: define o assunto onde está inserido o caso. Um exemplo seria "Projeto de Software".
- Tópico: procura definir um refinamento do tema. Um valor possível seria "Projeto Orientado a Objetos".
- Problema: este atributo tem o formato de um texto livre que deve ser preenchido com as características do caso, considerando o valor contido no atributo contexto. Para a recuperação, o texto pode conter palavras-chave.
- Solução: possui o mesmo formato do atributo problema, contudo com o sentido de armazenar a solução para o problema apresentado. Novamente, podem ser definidas palavras-chave.

A indexação dos casos utiliza inicialmente os quatro primeiros atributos. Esta organização interna permite uma rápida identificação dos casos relevantes a um assunto. O uso das palavras-chave permite um refinamento na recuperação dos casos. Este mesmo formato também é adotado para representar as experiências exploradas pelos estudantes em suas bases de crenças individuais. Neste caso, as palavras-chave podem também oferecer um recurso de avaliação do estudante e comparação com as bases de crenças de outros estudantes.

5.3.3 Agente Consultivo

O agente consultivo adota o paradigma de um assistente pessoal inteligente. Neste sentido, ele assiste o usuário através de uma monitoração constante, indicando a aplicação apropriada do conhecimento. Isto é feito através da verificação da situação do usuário e pela identificação dos casos mais apropriados para aquele momento.

Este agente pode ser consultado indiretamente pelo usuário, através do agente de interface. Neste caso, o usuário faz a solicitação ao agente de interface que faz o repasse ao agente consultivo. Por sua vez, o agente consultivo verifica as bases de crenças individuais de cada usuário e solicita ao agente de informação pelos casos mais relevantes na base de conhecimento global (quando comparados ao pedido do usuário).

Outra importante característica do agente consultivo é a avaliação das ações e construções das crenças do usuário. O agente procura encontrar similaridades entre os casos derivados de um usuário e os casos inicialmente contidos no material didático. Os resultados destas buscas podem ser disponibilizados ao usuário.

A estrutura do agente consultivo é praticamente a mesma utilizada para implementar o agente de informação (fig. 10). Novamente, não há a necessidade dos níveis de interface e interpretação. Entretanto, a diferença básica está no nível operacional. Enquanto o agente de informação possui facilidades de acesso ao conhecimento e banco de dados, o agente consultivo possui mecanismos para monitorar o usuário e para reconhecer situações onde é necessário apresentar explicações. Existe também a possibilidade de fornecer informações adicionais como exemplos ou contra-exemplos para o usuário.

O agente consultivo permite explorar duas abordagens diferentes: o ensino baseado em casos e ensino por demanda. Os benefícios pedagógicos destas abordagens foram discutidos no capítulo 2 e na seção 4.7 deste trabalho. O ensino baseado em casos é utilizado quando o agente procura apresentar situações aos estudantes, baseado na interpretação de suas crenças (base de casos local). Uma das formas mais valiosas de aprendizado ocorre quando o estudante com um problema descreve sua situação a um professor, então o professor é lembrado e responde através de uma história aplicável. Em tal situação, o estudante pode adaptar a história, enderecá-la e aplicá-la no seu problema. Esta parte do agente consultivo é baseada na estrutura da aplicação para ensino interativo SPIEL (*Story Producer for Interactive Learning*) (Burke e Kass, 1996).

O ensino baseado em demanda procura explorar o fato do estudante não receber toda a informação de uma única vez. O agente consultivo procura monitorar o estudante, identificando tópicos (pontos específicos) no material, e procura encontrar de forma autônoma um caso (experiência/explicação) adequado para aquele determinado tópico.

5.3.4 Arquitetura de Comunicação

Como apresentado anteriormente, a plataforma está estruturada com base na proposta de Genesereth e Ketchpel (1994), onde é definida uma arquitetura para interoperabilidade. A comunicação dos agentes ocorre através da troca de mensagens padronizadas que

adotam o padrão⁶ ACL (*Agent Communication Language*). Este padrão estabelece que a comunicação não ocorre diretamente entre os agentes, mas através de um programa especial denominado **Facilitador**. A figura 11 apresenta uma visão da arquitetura de comunicação adotada. São implementadas duas bases de dados locais: na primeira, o Facilitador armazena toda a informação necessária para executar o roteamento das mensagens. Na segunda base ele registra todas as mensagens trocadas. O formato destas bases é bastante simples e não cabe descrevê-las aqui.

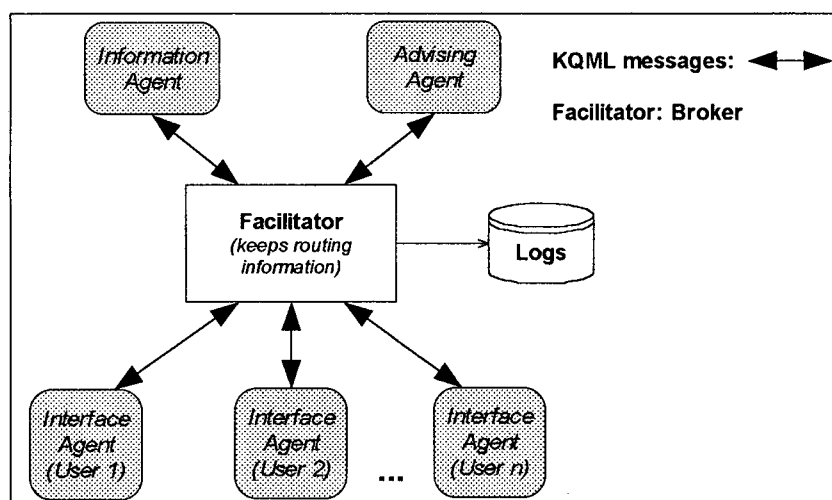


Fig. 11 – Arquitetura de comunicação adotada.

A principal diferença entre a arquitetura implementada e o padrão ACL está na quantidade de Facilitadores. No padrão ACL, é indicado que cada computador *host* possua um Facilitador para seus agentes locais. A arquitetura utilizada define apenas um único Facilitador para todos os agentes. Esta aparente limitação é, na verdade, uma decisão de projeto. Usualmente, os agentes de interface estarão espalhados em diferentes *hosts*, muitas vezes em diferentes redes locais. Em outras palavras, será muito difícil existirem dois agentes de interface sendo executados no mesmo computador. Esta avaliação fez com que a estrutura de comunicação fosse simplificada, reduzindo a troca de mensagens. Outra questão importante está no fato dos agentes nunca trocarem mensagens diretamente. No padrão ACL, um Facilitador pode passar o endereço de um agente a outro, permitindo que este entre em contato diretamente. A proposta adotada permite manter controle centralizado (neste caso, através do Facilitador) para facilitar a

⁶ A proposta ACL não é um padrão de direito, mas possui uma especificação bem documentada e consistente. Desta forma, este trabalho adota esta proposta como um padrão de fato.

formação de registros históricos. Estes registros são importantes para auxiliar no processo de avaliação dos usuários, uma vez que eles registram todo o processo de colaboração ocorrido entre os agentes.

A linguagem externa KQML está sendo utilizada para encapsular o conhecimento dos agentes. O padrão ACL adota a linguagem interna KIF para representar o conteúdo do conhecimento. A plataforma está preparada para aceitar esta linguagem, mas a versão atual não possui um tradutor KIF implementado. Entretanto, as mensagens KQML (*performatives*) podem carregar conhecimento em diversos tipos de linguagem como Prolog ou SQL.

As *performatives* KQML, apresentadas na figura 12, são utilizadas para realizar o processo inicial de conexão de um agente com o Facilitador.

```
(register-agent
  :sender      custom_agent
  :receiver    facilitator
  :language    s-expressions
  :content     (10.0.0.1 akInterface))

(reply
  :content     (custom_agent))
  :sender      facilitator
  :receiver    agent-12)

(unregister-agent
  :sender      agent-12
  :receiver    facilitator)
```

Fig. 12 – *Performatives* KQML para conexão inicial entre os agentes e o Facilitador.

A mensagem **register-agent** é enviada pelo agente ao Facilitador para solicitar seu registro no grupo. Os agentes consultivo e de informação também devem realizar esta operação, uma vez que o Facilitador não teria como saber as suas localizações na rede. No conteúdo desta mensagem são passados o número IP (*Internet Protocol*) e o tipo de agente. No exemplo dado, um agente de interface em um *host* com endereço "10.0.0.1" está solicitando seu registro.

A mensagem **unregister-agent** é enviada pelo agente ao Facilitador solicitando seu desligamento do grupo. Esta mensagem é usualmente feita por um agente de interface.

A mensagem *reply* é enviada pelo Facilitador ao agente que solicitou um registro. O conteúdo da resposta é um nome de agente disponível. Este nome não é conhecido pelo usuário, no caso do solicitante ser um agente de interface. O nome identifica o agente apenas no nível de comunicação. Este nível também recebe um número de porta (*socket*) temporário.

As *performatives* **register-agent** e **unregister-agent** não existem na especificação inicial da linguagem KQML (Finin et al, 1993). Entretanto, a linguagem permite a adição de novas *performatives*.

Depois do processo de registro concluído, um agente está pronto para receber e enviar mensagens através da plataforma.

5.3.5 Encapsulamento dos Casos

Todos os três tipos de agente definidos na plataforma (o Facilitador não está sendo considerado) devem possuir a habilidade de transmitir e receber casos da base de conhecimento. Desta forma, tornou-se necessário especificar como um caso poderia ser enviado através da rede. A solução foi encapsular os casos em mensagens KQML, uma vez que um caso é uma forma de representação do conhecimento.

Na primeira versão da plataforma, utilizava-se estruturas Prolog para a modelagem dos casos. Estas estruturas eram transportadas no parâmetro *content* de uma mensagem KQML. Entretanto, era necessário a utilização de um interpretador Prolog para que o caso pudesse ser explorado. Esta solução não era muito eficaz e necessitava da adoção de programas externos para funcionar adequadamente. Na versão atual, os casos estão modelados através de uma classe de objetos. Esta visão é interna à plataforma, uma vez que o banco de dados utilizado é relacional. Um importante aspecto desta nova abordagem é que novas classes podem ser generalizadas, permitindo que a representação dos casos possa ser melhorada em versões futuras. É importante notar que o uso de um banco de dados relacional limita a facilidade introduzida pela orientação a objetos. Entretanto, o modelo está parcialmente preparado para a migração para um banco de dados orientado a objetos.

Para o transporte adequado de um caso, a classe oferece um método de serialização que transforma um objeto caso em uma cadeia de caracteres (*string*). Para transmitir um caso através de uma mensagem KQML, o caso é convertido e colocado no parâmetro *content*, além do parâmetro *language* ser preenchido com o valor CASE-TEXT. Esta linguagem foi definida pela plataforma para indicar ao agente que o conteúdo da mensagem está carregando um caso.

5.4 Aplicação no Ensino à Distância

Uma grande parte dos sistemas de ensino baseados em computador utiliza metodologias tradicionais, onde o professor é o centro do conhecimento. Esta abordagem defende que o processo ensino-aprendizagem é somente uma simples transmissão de conhecimento. Desta forma, o aluno é convertido em um agente passivo com muito pouca participação na construção do conhecimento. Sistemas com estas características não conseguem explorar com muito sucesso as capacidades reais dos estudantes. Como apresentado, ferramentas que exploram a colaboração entre indivíduos podem alcançar resultados interessantes para o processo de aprendizagem.

Este trabalho utiliza a plataforma definida anteriormente para criar um ambiente de ensino colaborativo através da Internet. Neste contexto, agentes de interface podem interagir com os estudantes, os agentes de informação apresentarem o conhecimento sobre o tema estudado e os agentes consultivos podem ser interpretados como os monitores ou professores. O conceito de professor neste ambiente é aquele indivíduo que apresenta novas situações aos estudantes e avalia os resultados obtidos através da colaboração entre eles.

A plataforma definida na seção anterior pode ser ajustada perfeitamente como um ambiente de ensino interativo. Como visto nos sistemas de ensino baseados em computador, apresentados no capítulo 2, todos os componentes de um sistema tutor inteligente estão presentes:

- Módulo estudante: o agente de interface mantém a base de crenças individualizadas do estudante.

- Módulo do domínio: o agente de informação possui uma representação do conhecimento na forma de uma base de casos. As informações em formato de páginas HTML também fazem parte deste módulo.
- Módulo pedagógico: as metodologias de ensino empregadas pelo modelo são o ensino baseado em casos e ensino por demanda.
- Módulo especialista: o agente consultivo atua como um tutor inteligente, monitorando o conhecimento dos estudantes e oferecendo orientação durante o processo exploratório.
- Módulo de comunicação: apesar de parte deste módulo estar também presente no agente de interface, sua implementação foi construída sob a metáfora de um navegador Internet inteligente.

Torna-se importante ressaltar que as metodologias pedagógicas adotadas permitem o aprendizado por investigação. Sendo assim, a forma de aplicação da plataforma para o ensino pode ser vista como um sistema de ensino colaborativo.

5.4.1 Navegador Internet Inteligente

O módulo de comunicação necessita de uma metáfora que permita ao estudante acessar o sistema. A proposta é introduzir um ambiente na forma de um navegador Internet inteligente. A interação do navegador com a plataforma multi-agentes é feita através do agente de interface.

Uma das facilidades do navegador é sua habilidade de funcionar como uma ferramenta tradicional de treinamento através da Internet, permitindo que o estudante possa navegar livremente (de modo não linear) pelas informações. Contudo, o ambiente providencia também mecanismos que exploram a interação entre os estudantes. A figura 13 apresenta como o navegador é inserido na arquitetura de comunicação sem alterar a estrutura definida anteriormente na seção 5.3.4. O agente de interface passa a ser um componente do navegador inteligente, mantendo toda a responsabilidade de acessar a plataforma multi-agente.

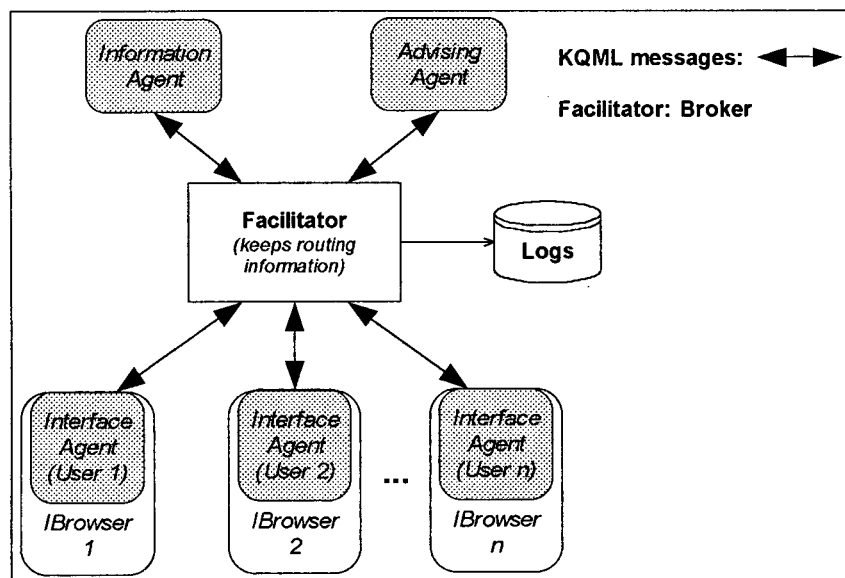


Fig. 13 – Arquitetura de comunicação com a inserção do navegador inteligente.

A interface proposta para o navegador inteligente possui quatro áreas básicas:

- **Área de navegação:** espaço na forma de um navegador para a Internet. Além de permitir o acesso a qualquer site na Internet, este navegador também permite abrir arquivos no formato HTML. Ele foi implementado de forma a armazenar todas as páginas navegadas em uma memória cache local. Esta característica permite acessar estas páginas, mesmo quando o estudante não estiver acessando a Internet. Existem opções para imprimir a página carregada e para trabalhar com servidores *proxy* (no caso do navegador estar sendo executado em alguma rede local com este serviço).
- **Área do assistente pessoal:** um personagem animado estabelece a metáfora para a representação do agente de interface e do agente consultivo. Este personagem possui uma série de animações pré-definidas que são exploradas pelo agente de interface para representar visualmente o que o agente está fazendo. Além disso, o personagem é capaz de conversar com o estudante através de mensagens escritas e faladas. Por sua vez, o estudante pode interagir com o personagem, solicitando informações adicionais como a recuperação de casos para um determinado tipo de problema.
- **Área de chamada da classe virtual:** esta área, localizada no canto superior esquerdo, apresenta os nomes de todos os estudantes que fazem parte da sala

de aula virtual. Os nomes podem aparecer em duas cores diferentes: verde para aqueles estudantes que estiverem com seu navegador em execução (considerando o acesso a Internet) e vermelho para os estudantes que também fazem parte do grupo, mas não estão acessando o ambiente naquele momento.

- **Área de conversação:** esta área está localizada no lado superior, entre a área de chamada e o personagem assistente. Seu objetivo é registrar todas as interações ocorridas entre o agente e o estudante. Desta forma, o estudante pode, a qualquer momento, revisar o que aconteceu. A informação desta área pode ser gravada pelo estudante na forma de um arquivo texto. Existe também a opção para limpar a área.

A figura 14 apresenta o navegador inteligente e suas áreas básicas. O personagem assistente assume o papel amigável do mágico "Merlin". Sua implementação utiliza uma interface com uma biblioteca oferecida pela Microsoft para a construção de assistentes (Microsoft, 1999). Esta biblioteca oferece ainda a possibilidade de desenvolver novos personagens com animações customizadas.

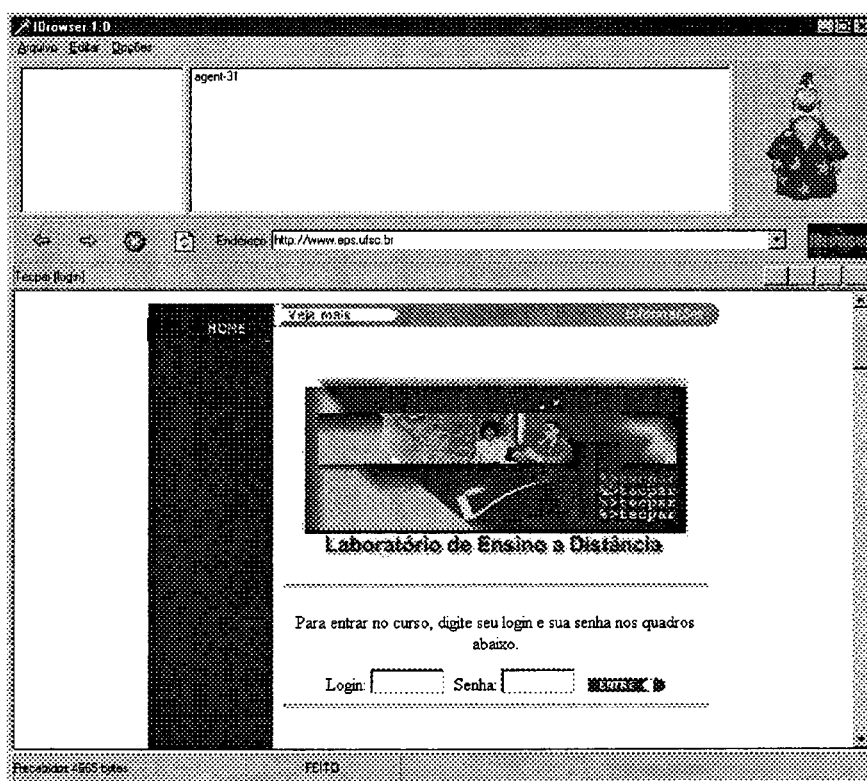


Fig. 14 – Ambiente colaborativo na forma de um navegador inteligente para a Internet.

5.4.2 Aplicação em Engenharia de Software

Para efeito de teste e avaliação do ambiente de ensino proposto, está sendo implantado um curso para a disciplina de Engenharia de Software. A figura 15 apresenta a interface do curso, onde podem ser observados três estudantes cadastrados.

Neste exemplo, pode-se verificar o resultado de uma solicitação do estudante para a definição do termo "classe". O agente de interface, após troca de mensagens com o agente consultivo, apresentou o caso recuperado. Por ser um caso do tipo "definição", não há informação no campo "problema". O campo "solução" é um texto livre que contém a definição de uma classe segundo a abordagem orientada a objetos. As palavras-chave estão indicadas pelos símbolos "[" e "]".

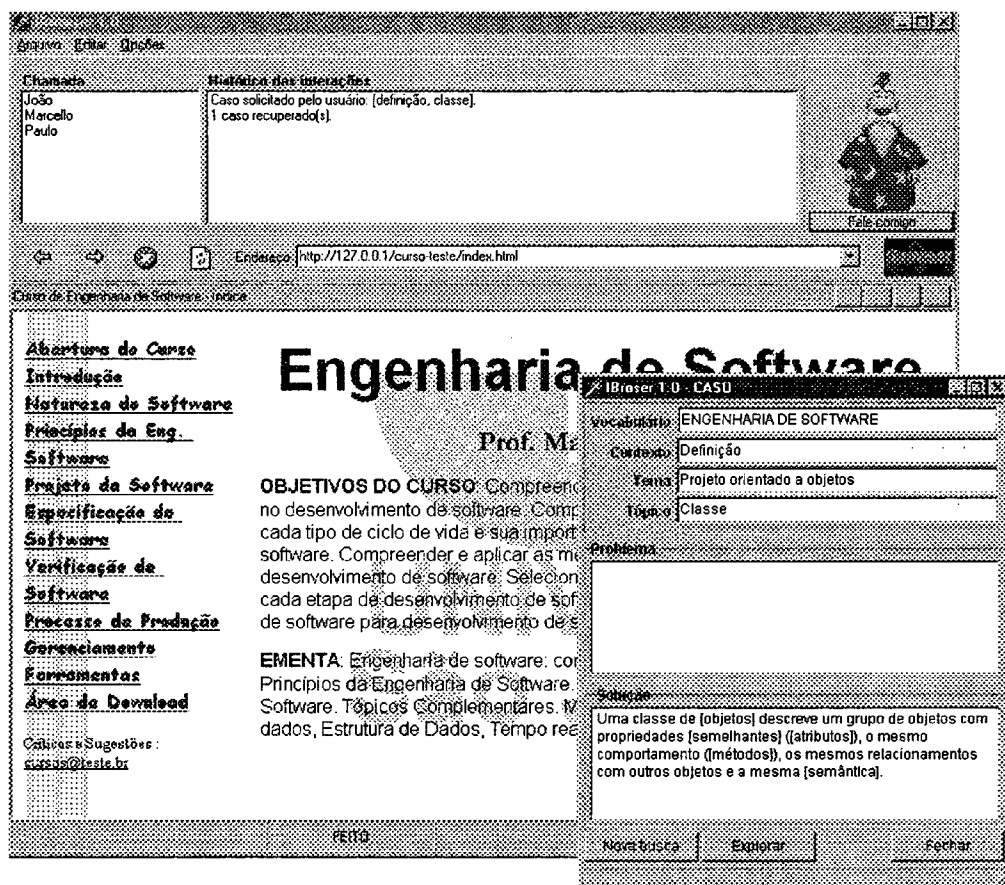


Fig. 15 – Recuperação de um caso.

O caso recuperado pode ainda ser explorado pelo estudante, o qual pode criar sua própria definição. Este processo de construção das crenças do estudante possibilita a individualização de cada aluno. A individualização considera que cada estudante é

diferente com diferentes níveis de conhecimento (Bork, 1997). Ao mesmo tempo, a colaboração entre alunos para resolver ou entender alguma situação providencia a aquisição de conhecimento.

6 Conclusões e Perspectivas Futuras

6.1 Conclusões

Este trabalho apresentou um modelo distribuído orientado a agentes para suporte ao trabalho e aprendizado colaborativo. Foi definida uma plataforma multi-agentes com diferentes classes de agentes, capazes de interagir entre si e com os seus usuários, permitindo facilidades de colaboração nos níveis agente-agente e agente-usuário.

A principal técnica de IA utilizada foi o raciocínio baseado em casos (RBC). Os princípios pedagógicos da abordagem RBC foram discutidos, procurando-se apresentar como eles foram incorporados à plataforma.

Este trabalho procura contribuir com um ambiente distribuído para suportar o ensino à distância através da Internet. O ensino à distância vem ganhando destaque como um importante instrumento de aprendizado e treinamento. Suas características permitem flexibilidade de tempo, local e horário aos seus participantes. Entretanto, grande parte dos sistemas de ensino através da Internet utilizam tecnologias para a transmissão de informação e não necessariamente de conhecimento. Desta forma, torna-se necessário a adoção de ferramentas inteligentes que incorporem novos princípios pedagógicos a este modelo de ensino. Neste sentido, a aplicação de assistentes pessoais e raciocínio baseado em casos oferece uma interessante abordagem para alcançar os objetivos do ensino.

A plataforma multi-agentes foi desenvolvida através de uma abordagem orientada a objetos, sendo possível o desenvolvimento de novas classes de agentes mais especializados. Estes novos agentes podem oferecer novas funcionalidades ao ambiente ou podem ser aplicados em outros domínios.

Uma importante característica observada no modelo para ensino colaborativo construído é a sua flexibilidade de utilização. O ambiente está preparado para aceitar vários cursos em diferentes domínios do conhecimento. Além disso, o processo colaborativo pode ser utilizado em outras aplicações que não o ensino. Um exemplo real de aplicação possível

está em sistemas de tomada de decisão em aspectos financeiros ou comerciais. A questão é explorar as similaridades de sistemas de ensino com os sistemas de apoio à tomada de decisão.

6.2 Perspectivas Futuras

Como visto, a arquitetura de agentes foi implementada, seguindo uma abordagem orientada a objetos, com uma estrutura de comunicação baseada no padrão ACL. Entretanto, os agentes não foram ainda habilitados com a capacidade de responder a todas as mensagens KQML necessárias para o funcionamento adequado da arquitetura. Neste sentido, torna-se necessário adicionar aos agentes a capacidade de analisar as situações de acordo com as mensagens KQML recebidas. Apenas as mensagens básicas de rede e algumas mensagens de consulta estão sendo tratadas.

A modelagem utilizada para a representação dos casos pode ser trabalhada para permitir um maior desempenho na recuperação de informações e para melhorar a capacidade de avaliar similaridades entre os casos. Esta nova modelagem pode adicionar a funcionalidade de representar um caso como vídeos sobre determinadas situações.

O navegador implementado é um ambiente que deve oferecer diversas ferramentas para explorar a colaboração e o aprendizado. Neste sentido, torna-se necessário adicionar facilidades como salas de discussão inteligentes e quadros gráficos compartilhados. Da mesma forma, poderia ser estudada a possibilidade de utilizar realidade virtual na área de navegação. Uma forma de alcançar rapidamente esta facilidade seria através implementação de páginas VRML (*Virtual Reality Markup Language*).

Outra questão importante a ser considerada são ferramentas para auxiliar na construção do material didático. Como visto, o ambiente está preparado para aceitar vários cursos. Entretanto, torna-se necessário montar a estrutura básica do curso e os casos que formarão o conhecimento global inicial. Este processo pode ser feito manualmente, mas exigiria um conhecimento desnecessário sobre a estrutura interna do ambiente por parte do grupo de professores.

7 Referências Bibliográficas

- AAMODT, A. e PLAZA, E., **Case-Based Reasoning: Foundational Issues, Methodological Variations, and Systems Approaches**. Artificial Intelligence Communications, Vol. 7, No. 1, 1994.
- ABOWD, G., ENGESLMA, J., GUADAGNO L. e OKON, O. **Architectural Analysis of Object Request Brokers**. Object Magazine, March, pp. 44-51, 98, 1996.
- AYALA, G. e YANO, Y. **Interacting With a Mediator Agent in a Collaborative Learning Environment**. Symbiosis of Human and Artifact: Furniture computing and design for human-computer interaction, Y. Anzai, K. Ogawa and H. Mori (eds.), Advances in Human Factors/Ergonomics, Elsevier Science Publishers, pp. 895-900, 1995.
- AYALA, G. e YANO, Y. **Intelligent Agents to Support the Effective Collaboration in a CSCL Environment**. Proceedings of the ED-TELECOM 96 World Conference on Educational Communications, Boston, Mass. AACE, Patricia Carlson and Fillia Makedon (eds.), pp.19-24, June, 1996.
- AZEVEDO, B. e TAVARES, O. **Um Sistema Tutor Inteligente para Suporte à Aprendizagem de "Conceitos de Orientação à Objetos"**. Revista Engenharia, UFES, Espírito Santo, Novembro, 1998.
- BECK, J., STERN, M. e HAUGSJAA, E. **Applications of AI in Education**. The ACM's First Electronic Publication, www.acm.org/crossroads/xrds3-1/aied.html, 1998.
- BOND, A. e GASSER, L. **An Analysis of Problems and Research in DAI**. Readings in Distributed Artificial Intelligence, Morgan Kaufmann, 1988.
- BORK, A. **The Future of Computers and Learning**. Technology Information/Cover Story, Technological Horizons in Education - THE Journal, 1997.

- BOURNE, J. R. **Asynchronous Learning Networks and World-Wide Engineering Education**. Engineering Education, Hewlett-Packard Company, Vol. 2, #1, Winter 1998.
- BOY, G. **Software Agents for Cooperative Learning**. In Software Agents, ed. J. M. Bradshaw, AAAI Press, 1997.
- BRADSHAW, J. **An Introduction to Software Agents**. In Software Agents, ed. J. M. Bradshaw, AAAI Press, 1997.
- BURKE, R. e KASS, A. **Retrieving Stories for Case-Based Teaching**. CBR: Experiences, Lessons & Future Directions, AAAI Press/MIT Press, David Leake editor, pp. 93-109, 1996.
- BYRNES, J. P. **Cognitive Development and Learning in Instructional Contexts**. Allyn & Bacon, 1996.
- CAGLAYAN, A., HARRISON C., **Agent Sourcebook**. John Wiley & Sons, 1997.
- CAGLAYAN, A., SONRRANSON. M., JACOBY, J., MAZZU, J., e JONES, R., **Lessons from Open Sesame!, a User Interface Learning Agent**. PAAM'96 Conference, pp. 61, 1996.
- CHAIB-DRAA, B. **Distributed Artificial intelligence: An Overview**. Encyclopedia of Computer Science and Technology. Kent, A. and Williams, J. (eds), 31 (suppl. 16), 1994. Reviewed on March 6, 1995.
- CROOK, C. **Educational Practice Within Two Local Computer Networks**. Computer Supported Collaborative Learning, Claire O'Malley (ed.), Springer-Verlag, pp. 165-182, 1994.
- D'AMICO, C., VICCARI, R., ALVAREZ, L. **A Framework for Teaching and Learning Environments**. VIII Simpósio Brasileiro de Informática na Educação, São José dos Campos, 1997.

- DURFEE, E. e ROSENSCHEIN, J. S. A. **Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples.** 13th International Workshop on Distributed Artificial Intelligence, Washington, July 17-19, 1994.
- FERGUNSON, W., BAREISS, R., BIRNBAUM, L. e OSGOOD, R. **ASK Systems: An Approach to the Realization of Story-Based Teachers.** Technical Report N. 22, Institute for the Learning Sciences, 1992.
- FININ, T., WEBER, J. et al. **DRAFT: Specification of KQML Agent Communication Language.** The DARPA Knowledge Sharing Initiative External Interfaces Working Group, <http://www.cs.umbc.edu/kqml/kqml.ps>, June 1993.
- FININ, T., FRITZSON, R., MCKAY, D. e MCENTIRE, R. **KQML as an Agent Communication Language.** Proceedings of the 3rd International Conference on Information and Knowledge Management, ACM Press, November, 1994.
- FISCHER, G. **Supporting Learning on Demand with Design Environments.** Proceedings of the International Conference on the Learning Sciences, Evanston, pp. 165-171 1991.
- FLEISCHHAUER, L. I. A. **O Uso da Tecnologia de Informação Baseada em Agente na Integração da Programação da Produção.** Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, 1996.
- FONER, L. **What is an Agent, Anyway? A Sociological Case Study.** Agents Memo, Media Lab, MIT, pp.93-101, 1993.
- FRANKLIN S. e GRAESSER, A. **Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents.** Proceedings of the Third International Workshop on Agent Theories, Springer-Verlag, 1996.
- GENESERETH, M. e FIKES, R. **Knowledge Interchange Format, Version 3.0 – Reference Manual.** Logic Group, Computer Science Department, Stanford University. June 1992.

- GENESERETH M. R. e KETCHPEL, S. P. **Software Agents**. Communications of the ACM, July 1994, pp. 48-53, 147.
- GENTNER, D. **Structure Mapping – A Theoretical Framework for Analogy**. Cognitive Science, Vol. 7, pp. 155-170, 1983.
- GENTNER, M. e FIKES, R. **Knowledge Interchange Format, Version 3.0 – Reference Manual**. Logic Group, Computer Science Department, Stanford University. June 1992.
- GOONATILAKE S. e KEBBAL, S. **Intelligent Hybrid Systems**. Intelligent Hybrid Systems, pp. 1-20, Wiley & Sons, England, 1995.
- HALL, P. e WOOD, P. **Intelligent Tutoring Systems: A Review for Beginners**. Canadian Journal of Educational Communication, 19 (2), pp. 107-123, 1990.
- HARRISSON C. G. e CHESS, D. M. and KERSHENBAUM, A. **Mobile Agents: Are they a good idea?** IBM Research Report. T. J. Watson Research Center, NY.
- HENDLER J. **Intelligent Agents: Where AI Meets Information Technology**. IEEE Expert Systems, December 1996, pp. 20-23.
- HEWITT, C. **Viewing Control Structures as Patterns of Passing Messages**. Artificial Intelligence 8, Vol. 3, 1977.
- KATZ, S. **Identifying the Support Needed in Computer-Supported Collaborative Learning Systems**, Proceedings of Computer Support for Collaborative Learning Conference, October 17-20, Indiana, USA, 1995.
- KAY, A. **Computer Software**. Scientific American 251 (3), pp. 53-59, September, 1984.
- KHAN, T. e YIP, Y. **Pedagogic Principles of Case-Based CAL**. Journal of Computer Assisted Learning, No. 12, pp. 172-192, 1996.

KHOO, N. e CHEN, J. **The Evolution of Intelligent Agent and Game Theory: Towards The Future of Intelligent Automation.** Surprise 95 Intelligent Agents and Games. Final Report, 1995.

KING D. e O'LEARY D. **Intelligent Executive Information Systems.** IEEE Expert Systems, December 1996, pp. 30-35.

KITANO H. **Nausicaä and the Sirens: A Tale of Two Intelligent Autonomous Agents.** IEEE Expert Systems, December 1996, pp. 60-61.

KNAPIK, M. e JOHNSON, J. **Developing Intelligent Agents for Distributed Systems.** Computing McGraw-Hill, NY:McGraw-Hill, 1998.

KOEDINGER, K. e ANDERSON, J. **Intelligent Tutoring Goes to School in the Big City.** Proceedings of 7th World Conference on Artificial Intelligence in Education, pp. 421-428, August 1995.

KOLODNER, J. **Reconstructive Memory, a Computer Model.** Cognitive Science, Vol. 7, pp. 281-328, 1983.

KOLODNER, J. **Case-Based Reasoning.** Morgan Kaufmann, 1993.

KOLODNER, J. e LEAKE, D. **A Tutorial Introduction to Case-Based Reasoning.** Case-Based Reasoning: Experiences, Lessons, & Future Directions. Ed. David B. Leake, AAAI Press, 1996.

LEAKE, D. B. **Case-Based Reasoning: Experiences, Lessons, & Future Directions.** AAAI Press, 1996.

LEJTER M. e DEAN T. **A Framework for the Development of Multiagent Architectures.** IEEE Expert Systems, December 1996, pp. 47-59.

LEMON, B., PYNADATH, D., TAYLOR, G. e WRAY, B. **Cognitive Architectures,** URL: <http://krusty.eecs.umich.edu/cogarch4/toc.html>, 1996.

MAES, P. **Agents that Reduce Work and Information Overload.** Communications of the ACM, Vol. 37, No. 7, pp.31-40, July 1994.

- MAES, P. **Intelligent Software: Programs That Can Act Independently Will Ease the Burdens that Computers Put on People.** IEEE Expert Systems, Vol. 11, No. 6, December 1996, pp.62-63,.
- MARIETTO, M., OMAR N., FERNANDES, C. **Tendências nas Áreas de Sistemas de Tutoria Inteligente e Modelagem do Aprendiz.** VIII Simpósio Brasileiro de Informática na Educação, São José dos Campos, 1997.
- MCARTHUR, D., LEWIS, M. e BISHAY, M. **The roles of artificial intelligence in education: current progress and future prospects.** RAND, Santa Monica, CA, USA, November, 1993.
- MERRIAM-WEBSTER. **The Merriam-Webster Dictionary.** Merriam-Webster Inc. Publishers, 1996.
- MICROSOFT, **Microsoft Agent Downloads.** www.microsoft.com/workshop/imedia/agent/, Janeiro, 1999.
- MINSKY, M. **IA Conversation With Marvin Minsky About Agents.** Communications of the ACM, July 1994, pp. 23-29.
- MINSKY, M. **A Framework for Representing Knowledge.** The Psychology of Computer Vision, P. Windston (ed.) McGraw Hill, 1975.
- MITROVIC, A. **A Knowledge-Based Teaching System for SQL.** Proceedings ED-MEDIA/ED-TELECOM'98, Freiburg, June 20-25, pp. 1027-1032, 1998.
- MOWBRAY, T. **Essentials of Object-Oriented Architecture.** Object Magazine, September 1995, pp. 28-32.
- NECHES, R., FIKES, R., FININ, T., GRUBER, T., PATIL, R., SENATOR, T. e SWARTOUT, W. **Enabling Technology for Knowledge Sharing.** AI Magazine, Vol. 3, No. 12, pp.36-56, 1991.
- NISSEN, M. **Intelligent Agents: A Technology and Business Application Analysis.** <http://haas.berkeley.edu/~heilmann/agents/#exnum>, November, 1995.

- NWANA, H. **Software Agents: An Overview**. Knowledge Engineering Review, Vol. 11, N. 3, pp. 1-40, September 1996.
- OLIVEIRA F. M. **Inteligência Artificial Distribuída**. IV Escola Regional de Informática, pp 54-73, Abril, 1996.
- OLIVER R., OMARI, A. e KNIBB, K. **ICreating Collaborative Computer-Based Learning Environments With The World Wide Web**. ASCILITE'97, December, 1997.
- OREY, M. e NELSON, W. **Development Principles for Intelligent Tutoring Systems: Integrating Cognitive Theory Into the Development of Computer-Based Instruction**. Educational Technology Research and Development, 41 (1), pp. 59-72, 1993.
- PATRIE C. **Agent-Based Engineering, The Web, and Intelligence**. IEEE Expert Systems, December 1996, pp. 24-29.
- PETERSON'S. **Guide to Distance Learning Programs**. Peterson's Publishing, Princeton, New Jersey, 1997.
- PLOETZNER R. e FEHSE, E. **Learning from Explanations: Extending One's Own Knowledge during Collaborative Problem Solving by Attempting to Understand Explanations Received from Others**. International Journal of Artificial Intelligence in Education, 9, 193-218, 1998.
- PORTER, L. **Creating the Virtual Classroom: Distance Learning with the Internet**. Wiley Computer Publishing, NY: Wiley & Sons., 1997.
- RIECKEN, D., P. M: **An Architecture of Integrated Agents**. Communications of the ACM, , Vol. 37, No. 7, pp. 107-116, 146, July 1994.
- RIESBECK, C. e SCHANK, R. **Inside Case-based Reasoning**. Lawrence Erlbaum Associates, Institute for the Learning Sciences, 1989.

- RUSSELL, S. e NORVIG, P. **Artificial Intelligence: A Modern Approach**. Englewood Cliffs, NJ: Prentice Hall, 1995.
- SCHERER, A. e SCHLAGETER, G. **CA Multi-Agent Approach for the Integration of Neural Networks and Expert Systems**. Intelligent Hybrid Systems, John Wiley & Sons, 1995.
- SCHANK, R. e ABELSON, R. **Scripts, Plans, Goals and Understanding: An Inquiry Into Human Knowledge Structures**. Lawrence Erlbaum Associates, Institute for the Learning Sciences, 1977.
- SCHANK, R. **Dynamic Memory: A Theory of Reminding and Learning in Computers and People**. Cambridge University Press, 1982.
- SCHANK, R. **Case-Based Teaching: Four Experiences in Educational Software Design**. Technical Report N. 7, Institute for the Learning Sciences, 1991.
- SCHANK, R., KASS, A. e RIESBECK, C. **Inside Case-based Explanation**. Lawrence Erlbaum Associates, Institute for the Learning Sciences, 1994.
- SHOHAM, Y. **Agent-oriented Programming**. Artificial Intelligence, 60, 51-92, 1993.
- SHOHAM, Y. **An Overview of Agent-Oriented Program**. In Software Agents, ed. J. M. Bradshaw, AAAI Press, 1997.
- SICHMAN, J., DEMAZEAU, Y. e BOISSIER, O. **When can Knowledge-based Systems be Called Agents**. IX Simpósio Brasileiro de Inteligência Artificial SBC, Rio de Janeiro, Brasil, 1992.
- SPECTOR, L. **Automatic Generation of Intelligent Agent Programs**. IEEE Expert Intelligent Systems, February 1997, pp. 3-4.
- SUN R. **Commonsense reasoning with rules, cases, and connectionist models: A paradigmatic comparison**. Fuzzy Sets and Systems, 82, 1996, pp. 187-200.
- SYCARA K. e PANNU A. et al. **Distributed Intelligent Agents**. IEEE Expert Intelligent Systems, December 1996, pp. 36-46.

- THIRY, M., BARCIA, R., e KHATOR, S. **Personal Assistance: An Intelligent Hybrid System Approach**. Industrial Engineering Research Conference 98, Alberta, Canada, May 1998.
- THIRY, M., BARCIA, R., KHATOR, S. e MARTINS A. **Intelligent Agent-based Approach for Distance Learning**. International Conference on Engineering Education 98, Rio de Janeiro, Brazil, August 1998.
- URBAN-LURAIN, M. **Intelligent Tutoring Systems: An Historic Review in the Context of the Development of Artificial Intelligence and Educational Psychology**, www.cse.msu.edu/~urban/ITS.htm, novembro, 1998.
- WATSON, I. **The Case for Case-Based Reasoning**. <http://www.salford.ac.uk/docs/depts/survey/staff/IWatson/ita01.html>, 1994.
- WATSON, I. **Applying Case-Based Reasoning: Techniques for Enterprise Systems**. Morgan Kaufmann, 1997.
- WEBBER, R. **Pesquisa Jurisprudencial Inteligente**. Tese de Doutorado, Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, 1998.
- WOOLDRIDGE, M. e JENNINGS, N. R. **Intelligent Agents: Theory and Practice**. Submitted to the Knowledge Engineering Review, 1994.
- WOOLDRIDGE, M. e JENNINGS, N. R. **Software Agents**. IEEE Review January 1996, pp. 17-20.
- WOOLF, B. **AI in Education**. Encyclopedia of Artificial Intelligence. Wiley & Sons, Inc., New York, 1992, pp. 434-444.